

Ingeniería Electrónica Industrial y Automática
(2018-2019)

Trabajo Fin de Grado

Diseño y desarrollo de expresiones faciales para un robot móvil

Ana Delgado Delgado

Tutora

Clara Gomez Blazquez

Leganés, Octubre 2019



Esta obra se encuentra sujeta a la licencia Creative Commons Reconocimiento – **No Comercial** – Sin Obra Derivada

RESUMEN

Uno de los principales objetivos de la robótica social es la inclusión de los robots en el día a día de las personas. Para conseguir este objetivo, es necesario que los robots sean percibidos como seres vivos, en lugar de máquinas.

La expresividad y el realismo favorecen la interacción humano-robot consiguiendo que los humanos perciban a los robots como seres inteligentes. Esta expresividad está basada en los diferentes rasgos faciales o movimientos involuntarios de los seres vivos.

El propósito de este trabajo es diseñar una cabeza y desarrollar un sistema que la dote de expresividad para un robot móvil perteneciente al laboratorio de robótica de la Universidad Carlos III de Madrid, con el fin, de favorecer la interacción entre robots y humanos. Este sistema debe ser compatible con el resto de la arquitectura del robot.

El sistema propuesto dota al robot de una funcionalidad que le permite expresar una serie de emociones según su estado interno. Para este proceso ha sido necesario recoger el marco teórico de conducta de un robot social más cercano a nuestro trabajo y existente hoy día en la psicología social.

Finalmente, tras el desarrollo del sistema, se realiza una serie de pruebas para comprobar que el diseño es adecuado y que se cumplen todos los objetivos propuestos.

ABSTRACT

One of the main objectives of social robotics is the inclusion of robots in people's daily lives. To achieve this goal, it is necessary that robots be perceived as living beings, rather than machines.

Expressiveness and realism favor human-robot interaction as it increases the probability that humans perceive robots as intelligent beings. This expressiveness is based on different facial features or involuntary movements of living beings.

The purpose of this work is to design a head and develop a system that provides expressiveness for a mobile robot belonging to the robotics laboratory of the Carlos III University in Madrid, in order to favor the interaction between robots and humans. This system must be compatible with the rest of the robot architecture.

The proposed system gives the robot a functionality that allows it to have a series of emotions according to its internal state. For this process it has been necessary to collect the theoretical framework of behavior of a social robot closer to our work and existing today in social psychology.

Finally, after the development of the system, a series of tests is carried out to verify that the design is adequate and all the proposed objectives are established.

DEDICATORIA

Quiero expresar mi agradecimiento a mi familia y amigos, por apoyarme en mis derrotas y por celebrar conmigo las victorias. Además de estar aguantándome cada día durante toda la etapa académica.

A mi tutora Clara Gómez por su orientación y dedicación en todo el proceso de la realización de este trabajo y por su paciencia.

ÍNDICE GENERAL

1. INTRODUCCIÓN.	1
1.1. Motivación	1
1.2. Objetivos	2
1.2.1. Objetivos específicos.	2
1.3. Organización del documento	3
1.4. Marco regulador	5
2. ESTADO DEL ARTE.	7
2.1. Antecedentes de los robots sociales	7
2.2. Fisionomía de los robots sociales.	11
2.3. Simulación de emociones en robots sociales	15
3. DISEÑO CONCEPTUAL DEL SISTEMA.	19
3.1. Diseño del rostro del robot	19
3.2. Diseño de la carcasa de la cabeza.	21
4. DESARROLLO DE LA EXPRESIÓN VISUAL DEL ROSTRO DEL ROBOT .	23
4.1. Ingeniería de requerimientos para el desarrollo electrónico	23
4.1.1. Arduino	24
4.1.2. Módulo de Pantalla Táctil TFT LCD 800x480	25
4.1.3. Tarjeta de expansión Escudo LCD táctil para Arduino Mega	26
4.1.4. Tarjeta de Memoria SD de 2 GB	27
4.2. Ingeniería de requerimientos para el desarrollo <i>software</i>	27
4.2.1. Robot Operating System.	27
4.2.2. Arduino	29
4.2.3. Inkscape.	30

4.3. Integración ROS/ARDUINO	31
4.4. Expresiones implementadas	32
4.5. Implementación del programa	34
4.5.1. Implementación del Subscriber	36
4.5.2. Implementación del estado de parpadeo y dormido	37
4.5.3. Implementación de las imágenes	38
4.5.4. Implementación tarjeta SD	40
4.5.5. Verificación código.	40
5. DISEÑO DE LA CARCASA DE LA CABEZA	42
5.1. Ingeniería de requerimientos para el desarrollo mecánico	42
5.1.1. FREECAD	43
5.2. DISEÑO CON FREECAD	43
5.3. Diseño de la cabeza	44
5.3.1. Cabeza versión 1	45
5.3.2. Cabeza versión 2	46
5.4. Diseño de la base.	48
5.5. Unión de los diseños	49
6. CONSTRUCCIÓN Y PRUEBAS	50
6.1. Construcción	50
6.2. Pruebas	53
6.2.1. Prueba de reacción del sistema	53
6.2.2. Prueba del estado de parpadeo	54
6.2.3. Prueba de refresco de las diferentes expresiones	55
6.2.4. Evaluación expresiones	56

7. DISCUSIÓN DE LOS RESULTADOS	59
7.1. Conclusiones	59
7.2. Trabajos futuros	60
8. ENTORNO SOCIO ECONÓMICO	62
8.1. Entorno socio económico	62
8.2. Presupuesto	64
9. METODOLOGÍA Y PLAN DE TRABAJO	66
BIBLIOGRAFÍA	67

ÍNDICE DE FIGURAS

2.1	a) Robot tortuga de Walter después de la restauración. Fuente: [19]. b) Anatomía de una réplica de las tortugas robot de Walter. Fuente: [20]. . .	8
2.2	Robot de investigación URBANO. Fuente: [25].	10
2.3	Robot de entretenimiento Aibo. Fuente: [28]	10
2.4	Robot terapéutico KEEPON. Fuente: [29]	11
2.5	Robot de asistencia en el hogar Care-O-bot. Fuente: [31].	11
2.6	Versión simplificada del gráfico del Uccany Valley que aparece en el artículo de Energy original de Mori (1970).	12
2.7	Resultados de la investigación realizada por la Universidad de California, en San Diego [33]. Respuesta cerebral ante interacción: R) Robots, A) Humanoide, H) Humano	13
2.8	Evolución de la fisonomía desde el robot Qrio hasta el androide Philip K. Dick, creado a la semejanza del propio Philip K. Dick . Fuente : [36]. . .	15
2.9	Expresiones faciales universales propuestas por Ekman (1972). Fuente: [42].	16
2.10	Diferentes expresiones planteadas para el rostro del robot.	18
3.1	Diagrama de flujo explicando de manera conceptual el programa de la representación de la cara del robot.	20
3.2	a) Diseño de la carcasa de la cabeza mediante el programa FreeCad. b) Diseño en Freecad de la base que une la cabeza con el cuerpo del robot. .	21
4.1	Interconexionado del sistema electrónico: a) Vista lateral donde se muestra la adaptación de la pantalla y el Arduino mediante la tarjeta de expansión. b) Vista inferior donde se puede ver la conexión de la tarjeta SD.	24

4.2	Arduino MEGA 2560	25
4.3	Pantalla TFT LCD 800x480	25
4.4	Tarjeta de expansión Escudo LCD	26
4.5	Esquema de como conectar la pantalla LCD con el escudo para su correcto funcionamiento	26
4.6	Tarjeta de Memoria SD Transcend de 2 GB.	27
4.7	Ejemplo de red P2P [46].	28
4.8	Estructura de datos <i>geometry_msgs/Twist.msg</i>	29
4.9	Ejemplo comunicación ROS. Tanto usando <i>topics</i> (flujo inferior) como servicios (flujo superior)	29
4.10	Pantalla de herramientas del programa Arduino	30
4.11	Diagrama explicativo de la integración ROS/Arduino.	31
4.12	División en píxeles de las diferentes partes que constituyen la cara del robot.	32
4.13	Transición de las diferentes expresiones del robot. a) Neutral b) Triste c) Feliz d) Asustado e) Parpadeo f) Muy asustado g) Dormido.	33
4.14	Diagrama del funcionamiento de la función <i>Callback_Destino</i>	37
4.15	Diagrama del funcionamiento de la función <i>Callback_Velocidad</i>	38
4.16	Gráfica del tiempo de refresco de la imagen según le agregamos rasgos faciales	40
5.1	Lion pro 3D	43
5.2	Primera versión de la cabeza del robot.	45
5.3	Rectángulo circunscrito	46
5.4	Segunda versión de la cabeza del robot	47
5.5	Segunda versión de la cabeza del robot	47
5.6	Vistas de la base de la cabeza.	48
5.7	Diseño de la base que une la cabeza con el cuerpo del robot.	49

5.8	Diseño de la cabeza del robot uniendo ambas partes.	49
6.1	Modelo impreso en 3D de la base.	50
6.2	Modelo impreso en 3D de la cabeza sobre la base.	51
6.3	Imagen en detalle de los taladros y la unión de la base con la cabeza. . . .	51
6.4	Diagrama de conexión del sistema de expresión visual	52
6.5	Robot presentado en este trabajo tras llevar a cabo el montaje de los dispositivos electrónicos y pintado	52
6.6	Expresión del estado neutral.	56
6.7	Expresión del estado feliz.	57
6.8	Expresión del estado triste.	57
6.9	a)Expresión del estado asustado. b) Expresion del estado muy asustado. .	57
6.10	Expresión del estado dormido.	58
8.1	Países con alto riesgo de sustitución de empleados por autómatas. Fuente: [52]	63
8.2	Costes de personal	64
8.3	Costes de software	64
8.4	Costes de hardware	65
8.5	Costes total	65
9.1	Diagrama de Gantt del proyecto	66

ÍNDICE DE TABLAS

2.1	Interpretación en función de la posición de las cejas	16
4.1	Definición de los diferentes parámetros del constructor de la librería UTFT.	39
4.2	Definición de los diferentes parámetros de la herramienta de dibujo.	39
6.1	Tiempo en reaccionar del sistema	53
6.2	Tiempo de parpadeo y naturalidad que transmite	54
6.3	Tiempo de refresco de las diferentes expresiones	55

1. INTRODUCCIÓN

En este capítulo se va a realizar una introducción al trabajo que vamos a presentar. En primer lugar, en la sección 1.1 se explica la motivación principal del trabajo, con una breve introducción de la evolución de la robótica social y sus características. A continuación, en la sección 1.2 se exponen los objetivos del trabajo, y por último, en la sección 1.3 se presenta cómo se conforma la estructura del documento.

1.1. Motivación

La robótica se puede definir como la ciencia encargada del diseño, la fabricación y la programación de los robots, según el Diccionario de la Real Academia Española [1]. En estas últimas décadas la robótica ha conseguido una gran cantidad de avances, se ha evolucionado desde unos autómatas bastantes simples, que son capaces de realizar pocas funciones, a unas máquinas realmente complejas, que son capaces de imitar actitudes humanas y tener una interacción humano-robot.

Dentro de los tipos de robótica, la robótica social es una de las que se abre paso día a día en campos como la medicina, la industria, el entretenimiento y la asistencia social. La robótica social se encarga de buscar la interacción humano-robot, siguiendo comportamientos y normas sociales. Estas habilidades comunicativas dependen del avance de la inteligencia artificial, para poder así, garantizar una interacción total entre el humano y el robot [2]. Estos robots deben de cumplir una serie de requisitos [3] : deben de ser autónomos, tener una apariencia amigable para el ser humano y poseer, mediante una programación sencilla, la capacidad de aprendizaje.

Se han realizado varios artículos sobre este campo de la robótica social [4] [5] que demuestran que los humanos interaccionan de manera más favorable con los robots cuando estos tienen un aspecto realista y están dotados una cierta expresividad. Para conseguir esta interacción es necesario el desarrollo de sistemas el estudio de dispositivos capaces de percibir, medir e interpretar las emociones humanas gracias a la computación afectiva [6]. Esto supone un avance en la inteligencia artificial, ya que hace posible que los robots

sociales lleguen a convivir con los humanos, adaptándose a sus necesidades y estados de ánimo. Esto es necesario debido a que los seres humanos están rodeados de emociones con las que se comunican. Por ello, resulta poco natural comunicarse con un robot que no es capaz de expresar sentimientos. La computación afectiva pretende mejorar la interacción humano-robot haciéndola más natural en este sentido.

Teniendo en cuenta los aspectos descritos anteriormente, en este trabajo de fin de grado surge la necesidad de implementar un sistema de expresividad en un robot móvil con plataforma capaz de variar su altura, y perteneciente al laboratorio de Robótica de la Universidad Carlos III de Madrid (UC3M). El fin de este trabajo es que dicho robot sea agradable al trato con humanos y de que éstos lo perciban como un ser inteligente, ya que se pretende que en un futuro este robot pueda entrar en el ámbito de los servicios y realizar tareas de ayuda a los humanos.

1.2. Objetivos

Los objetivos principales de este proyecto son, en primer lugar, la implementación, y sus pruebas, de un programa encargado de mostrar las diferentes expresiones según el estado del robot. En segundo lugar, el diseño de la carcasa de la cabeza y la elección del hardware utilizado.

Toda la programación existente en el proyecto se lleva a cabo sobre el sistema ROS, ya que éste es el utilizado por el robot del laboratorio dada su versatilidad. A su vez, el lenguaje de programación empleado en el desarrollo del código para la implementación de la pantalla es Arduino.

A continuación, se muestra el desglose de este objetivo principal.

1.2.1. Objetivos específicos

Para lograr el objetivo principal se han estipulado una serie de objetivos secundarios o específicos que deben ser cumplidos:

- Analizar las plataformas de interacción humano-robot más significativas que existen en la actualidad.

- Estudiar los principales aspectos de la disciplina de la psicología social que son relevantes en la informática y, más concretamente, en el campo de la interacción humano-robot.
- Analizar los elementos de la psicología social que podrían ser implementados para crear el proceso de interacción lo más natural posible.
- Buscar el hardware necesario para poder desarrollar la plataforma robótica, cumpliendo con las necesidades demandadas, y justificar el empleo de cada componente.
- Creación de un sistema encargado de controlar la expresividad de la cara del robot.
- Desarrollar la compatibilidad del sistema de expresividad del rostro del robot con el resto de las funcionalidades del cuerpo del robot.
- Proponer un diseño para la cabeza del robot que incluya, además de ésta, la base que la unirá con el resto del cuerpo.
- Creación de una estructura con una impresora 3D para el montaje de los diferentes dispositivos

El desarrollo de estos objetivos específicos configura este trabajo fin de grado, con los que se hará posible la construcción de un robot capaz de tener una interacción humano-robot sustentada por las principales teorías desarrolladas en el ámbito de la psicología social.

1.3. Organización del documento

El presente documento consta de una primera parte que incluye: un resumen, tanto en castellano como en inglés, agradecimientos, el índice de contenidos y un índice de figuras y de tablas.

La segunda parte del documento está dividida en 9 capítulos, que serán detallados a continuación:

- **Capítulo 1:** contiene una breve introducción a la robótica social y a los diferentes estudios realizados sobre la interacción humano-robot. Además, se exponen los objetivos que debe cumplir el proyecto, así como su estructura y el marco regulador en el que se encuentra.
- **Capítulo 2:** presenta una descripción del estado del arte, explicando los antecedentes y realizando un estudio de la fisionomía de los robots sociales para lograr una interacción humano-robot adecuada.
- **Capítulo 3:** se describe de manera conceptual el diseño del rostro del robot y de la carcasa de la cabeza.
- **Capítulo 4:** se detalla el sistema de expresión visual del rostro del robot y la ingeniería de requerimientos para que el desarrollo cumpla con los objetivos. Por último, se explica la adaptación para la integración con el resto del cuerpo robot.
- **Capítulo 5:** se diseña la carcasa de la cabeza y de la base que la une al resto del cuerpo robot mediante el programa FreeCad. Además, se mostrará el diseño completo con todos los elementos y componentes ya ensamblados.
- **Capítulo 6:** se explica cómo se ha realizado el montaje de las piezas diseñadas para la cabeza, para después evaluar el sistema de expresividad en cada uno de los estados internos del robot, así como analizar los resultados obtenidos.
- **Capítulo 7:** se realiza una breve conclusión sobre lo que supone la realización del presente proyecto, así como el camino que queda abierto para futuros trabajos.
- **Capítulo 8:** se evalúa el impacto socio-económico del sistema y se incluye un presupuesto detallando el coste del proyecto.
- **Capítulo 9:** se expone el cronograma del plan de trabajo seguido para la realización del proyecto.

Al final del proyecto se incluye la bibliografía y los anexos correspondientes.

1.4. Marco regulador

Esta sección se incluye con el propósito de realizar un breve análisis de la legislación vigente que afecta a este proyecto de forma directa. Es decir, que afecta a la propiedad intelectual y al marco actual de responsabilidad civil en la robótica.

La asignatura de Trabajo Fin de Grado está regulada a nivel estatal en el **Real Decreto 1393/2007**, por el que se establecen la ordenación de las enseñanzas universitarias de oficiales, modificado por RD 99/2011 (Doctorado) y por RD 43/2015 ¹.

El sistema de expresividad propuesto en este trabajo cumple con la **Ley de propiedad intelectual (LPI)** establecida en el **R.D.L. 2/2018** ². Al tratarse de una obra de investigación se acoge al “**derecho de cita**”, que permite hacer referencia a obras con copyright previamente divulgadas, indicando adecuadamente el autor y la fuente [7].

En este trabajo de fin de grado se va a trabajar con un robot social, por lo que es fundamental conocer la normativa aplicada al campo de la robótica en España. El estado actual de esta regulación se recoge en el artículo [8]. Además, este proyecto está financiado por la Universidad Carlos III de Madrid, que a su vez recibe financiación para la investigación por parte del Estado, regulada por la **Orden ECC/1333/2015, de 2 de julio**³.

El *software* del robot al que le incorporaremos la capacidad de interacción se encuentra bajo una licencia privativa gestionada por un grupo de investigación de la propia Universidad Carlos III de Madrid, por lo que no será necesaria la aplicación de la **Ley de patentes (LPa 24/2015)** ⁴para regularlo.

Para el desarrollo de este proyecto se han utilizado diversas herramientas informáticas,

¹BOE núm. 260, de 30 de octubre de 2007.

²Real Decreto-ley 2/2018, de 13 de abril, por el que se modifica el texto refundido de la Ley de Propiedad Intelectual, aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril, y por el que se incorporan al ordenamiento jurídico español la Directiva 2014/26/UE del Parlamento Europeo y del Consejo, de 26 de febrero de 2014, y la Directiva (UE) 2017/1564 del Parlamento Europeo y del Consejo, de 13 de septiembre de 2017 (BOE núm.91, de 14 de abril de 2018).

³Ministerio de Economía y Competitividad de España. Orden ecc/1333/2015, de 2 de julio, por la que se aprueban las bases reguladoras para la concesión de ayudas públicas del programa estatal de liderazgo empresarial en idi, en el marco del plan estatal de investigación científica y técnica y de innovación 2013-2016., 2015.

⁴ Jefatura del Estado de España. Ley 24/2015, de 24 de julio, de patentes, 2015.

sujetas a sus correspondientes licencias:

- El sistema operativo Ubuntu, sujeto a licencia **GNU GPL** ⁵, que garantiza a los usuarios la libertad de usar, estudiar, compartir y modificar el *software* [9].
- El framework ROS, un *software* libre bajo términos de licencia **BSD** ⁶, que permite libertad para uso comercial e investigador [10][11].
- El desarrollador de *software* y *hardware* de fuente abierta Arduino, que bajo la Licencia Pública General de **GNU (GPL)** y la Licencia Pública General Reducida de **GNU (LGPL)** [12] permite la manufactura de las placas Arduino y la distribución del *software*.
- La aplicación libre FreeCAD de diseño asistido por computadora libre bajo términos de la Licencia Pública General de **GNU (GPL)** y la Licencia Pública General Reducida de **GNU (LGPL)**.

El sistema desarrollado consiste en un robot diseñado para la interacción con humanos, sin embargo, el sistema no recoge datos ni información personal, por lo que no es aplicable la **LOPD 3/2018** ⁷, que garantiza la protección de los datos personales de las personas físicas.

Por último, se cumplen los requisitos del Reglamento de organización y funcionamiento del Comité de Ética en Investigación de la Universidad Carlos III de Madrid ⁸, en el que se establecen las pautas de actuación, recomendaciones y compromisos para la realización de las actividades de investigación en la Universidad [13].

⁵GNU General Public License

⁶Berkeley *software* Distribution

⁷Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (BOE núm.294, de 6 de diciembre de 2018).

⁸Aprobado por la Universidad Carlos III de Madrid, el 27 de Abril de 2017.

2. ESTADO DEL ARTE

En este capítulo se va a presentar el estudio del estado del arte en el campo de la interacción humano-robot abordando, en primer lugar, la historia de los robots sociales, definiendo qué es un robot social y sus clasificaciones. A continuación, nos centramos en la fisionomía de los robots sociales ayudándonos de diferentes estudios, entre los que destaca el realizado por Masahiro Mori [14]. Por último, se explica la representación de emociones en los robots sociales y los requisitos que éstos deben tener para poder expresarse de una manera reconocible al ser humano.

2.1. Antecedentes de los robots sociales

La interacción Humano-Robot (HRI) es uno de los campos de la robótica de mayor actualidad e interés. La llamada robótica social es la rama de la robótica que se encarga de integrar y hacer que los robots posean comportamientos empáticos hacia los humanos, esto se consigue añadiendo características antropomórficas o zoomórficas físicas. De este modo, los robots que realizan tareas como proporcionar información del ambiente que les rodea o rastrear la ubicación de una persona resultan más cercanos y son más aceptados por los humanos con los que interactúan.

Entre las propuestas que explican como se definen los robots sociales encontramos la de Dautenhahn y Billard [15], quienes definen a los robots sociales como agentes que forman parte de un grupo heterogéneo: una sociedad de robots y humanos. Según ellos, son capaces de reconocerse entre sí, participar en interacciones sociales, percibir e interpretar el mundo en términos de su propia experiencia y comunicarse de forma explícita aprendiendo unos de otros.

Todo robot social consta de cuatro elementos esenciales [16], siendo los tres primeros comunes a todos los robots y el último el que diferencia al robot social del resto de tipos:

1. **La parte física:** es la estructura del robot, sobre la cual se encuentran los diferentes elementos como bastidores y mecanismos necesarios.

2. **La parte reactiva:** está conformada por los sensores y actuadores del robot.
3. **La parte deliberativa:** en ella se enmarca el grado de decisión del robot.
4. **La parte social:** es la encargada de la interacción humano-robot. Para esta interacción, el robot posee un modelo cognitivo-afectivo, es decir, capacidades como la comunicación, la comprensión y el aprendizaje.

La robótica social se inicia con las investigaciones sobre las comunidades biológicas animales [17], con el objetivo de trasladar a los robots las interacciones entre los individuos que las componen.

Este origen de la robótica social se le atribuye a William Grey [18] que en los años 40 creó los primeros robots inspirados biológicamente en tortugas (figura 2.1). Estos robots llamados Elmer y Elsie, fueron los primeros robots autónomos y móviles. Gracias a las tres ruedas que constituían la función del movimiento (una frontal giratoria y dos traseras fijas) podían moverse con soltura evitando los obstáculos que se les fueran apareciendo en su camino y después volver a su sitio de carga, de esta manera funciona de una forma autónoma. Además, mediante faros ensamblados en la parte frontal de la carcasa, interactuaban de una forma similar a la social, a pesar de la falta de comunicación explícita o reconocimiento mutuo. Esta interacción se basaba en la fototaxia positiva que incluían los faros y a su capacidad para realizar movimientos orientados en respuesta a la intensidad de la luz, dirigiéndose hacia el haz de luz.

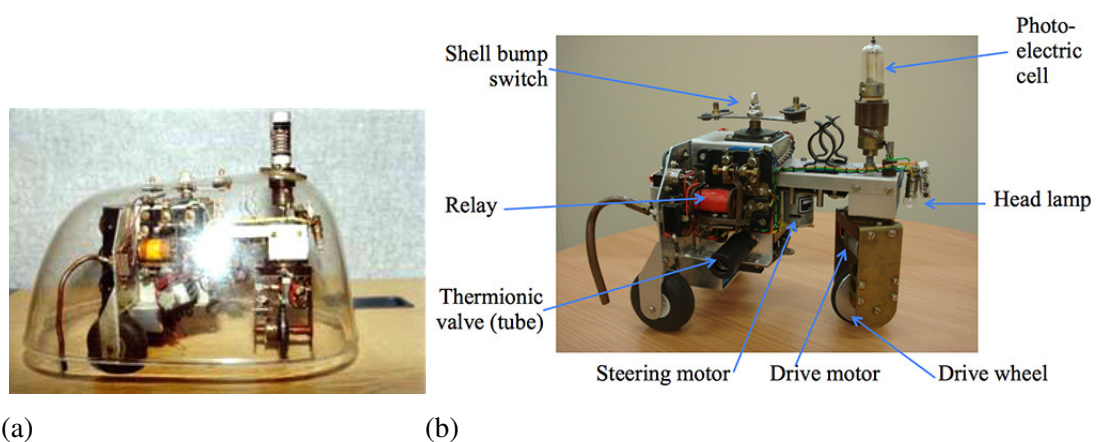


Fig. 2.1. a) Robot tortuga de Walter después de la restauración. Fuente: [19]. b) Anatomía de una réplica de las tortugas robot de Walter. Fuente: [20].

La clasificación de los robots, aunque se base en elementos técnicos como la arquitectura, el diseño de la interfaz y el *software*, también se puede realizar según cómo el robot se enfrenta al modelo social en el que tiene que trabajar y a la complejidad de la interacción con el escenario que debe llevar a cabo. Como establece Breazeal en su artículo [21] se clasifican en cuatro clases:

- **Socialmente evocador:** robots que imitan costumbres de las personas para tener una apariencia más humana.
- **Interfaz social:** robots que proporcionan una interfaz natural mediante señales sociales humanas. Es decir, aquellos que se comunican de un modo humano para facilitar la interacción con las personas ayudándose de gestos, expresiones, etc.
- **Socialmente receptivo:** robots socialmente pasivos que responden a los esfuerzos que las personas realizan para interactuar con ellos, sin embargo, no comprometen activamente a las personas para satisfacer sus objetivos. Por otro lado, pueden beneficiarse de la interacción a través de habilidades de aprendizaje por imitación.
- **Sociables:** robots que actúan activamente con los humanos para satisfacer sus necesidades. Estos robots tratan de involucrar a las personas de una manera social no solo para el beneficio de las mismas, sino también para beneficiarse ellos mismos, por ejemplo, para aprender.

Teniendo en cuenta la clasificación anterior, hay que destacar que también es posible diferenciar los robots según la aplicación para la que fueron creados, como plantea D. Parisi [22], donde explica que los robots sociales se pueden dividir como:

- **Robots sociales para la investigación.** Un ejemplo de este tipo de robots es URBANO [23], este proyecto forma parte del grupo de Control Inteligente del Centro de Automática y Robótica de la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid. URBANO es un robot utilizado para investigar en distintos campos de la robótica social desde la navegación autónoma [24] hasta modelos emocionales como el de nuestro trabajo. URBANO (figura 2.2) consta de una plataforma con ruedas que habilita el movimiento, una cara capaz de mostrar expresiones y un brazo que permite realizar gestos.



Fig. 2.2. Robot de investigación URBANO. Fuente: [25].

- **Robots sociales para el entretenimiento.** Este tipo de robots sociales constituyen un campo de investigación en auge. Según estima F. Kaplan [26] estos robots pasarán a formar parte la vida cotidiana de las familias en los próximos años, ya sea como mascotas, juegos o a través de *hobbies*. Como se ve en la figura 2.3, Aibo [27] es un ejemplo de estos robots sociales como mascotas. Aibo es capaz de imitar a un perro gracias a su capacidad para reconocer a su dueño, así como obedecer órdenes de voz, también puede ver gracias a la cámara que lleva instalada en su nariz y que le permite reconocer el entorno. Con esta información y su sistema de inteligencia artificial, va aprendiendo y creando una propia "personalidad".



Fig. 2.3. Robot de entretenimiento Aibo. Fuente: [28] .

- **Robots sociales educativos.** Dentro de esta clasificación podemos encontrar robots dedicados a educar de una forma específica, por ejemplo, en el caso de estudiantes con necesidades especiales. KEEPON [29], es un robot social terapéutico que apoya el tratamiento de pacientes con trastornos del espectro autista, la finalidad de este robot es tener una participación positiva con las personas que tienen una conducta social abrupta (ver figura 2.4). KEEPON funciona de tal manera que el terapeuta va

controlando el robot para facilitar la interacción social con el paciente. También es una herramienta de grabación en el consultorio, laboratorio o sala de juegos.



Fig. 2.4. Robot terapéutico KEEPON. Fuente: [29]

- **Robots sociales de apoyo.** Estos robots tienen como objetivo ayudar a las personas en las actividades cotidianas [30], especialmente a aquellas que no pueden valerse por sí mismas. Dentro de este grupo podemos diferenciar entre robots de ayuda y cuidados, como puede verse en la figura 2.5, y robots para facilitar la movilidad.



Fig. 2.5. Robot de asistencia en el hogar Care-O-bot. Fuente: [31].

2.2. Fisionomía de los robots sociales

Dentro de la robótica social, un factor muy importante es la gesticulación. Se debe tener en cuenta que la capacidad para interactuar con humanos depende de que tenga un aspecto amigable, de manera que el usuario que está interactuando con él se sienta lo más cómodo posible.

Según D.A.B Loza y S.M Popoli [32], lo más común es pensar que cuanto más se parezca un robot al ser humano, más cómodos nos sentiremos al interactuar con él. Sin embargo, el experto en Robótica Masahiro Mori [14], demostró que esto no era así, contrastando esta hipótesis en distintos trabajos. Sus resultados se representan con el gráfico de la figura 2.6, donde se muestra la evolución de los diseños de los robots en función del parecido humano. Como resultante se obtiene el agrado del diseño para las personas, dicho gráfico es conocido como *Uncanny Valley*. Este concepto expresa mediante una función matemática la aceptación de un robot en función de su diseño. En esta gráfica tomamos como referencia dos ejemplos de robot, el primero es un robot industrial, que se caracteriza por tener tres o más ejes diseñados para la ejecución de procesos industriales. Por otro lado, el robot humanoide, que es un robot diseñado para simular el movimiento y forma del ser humano, consta normalmente de un torso, una cabeza, dos brazos y dos piernas.

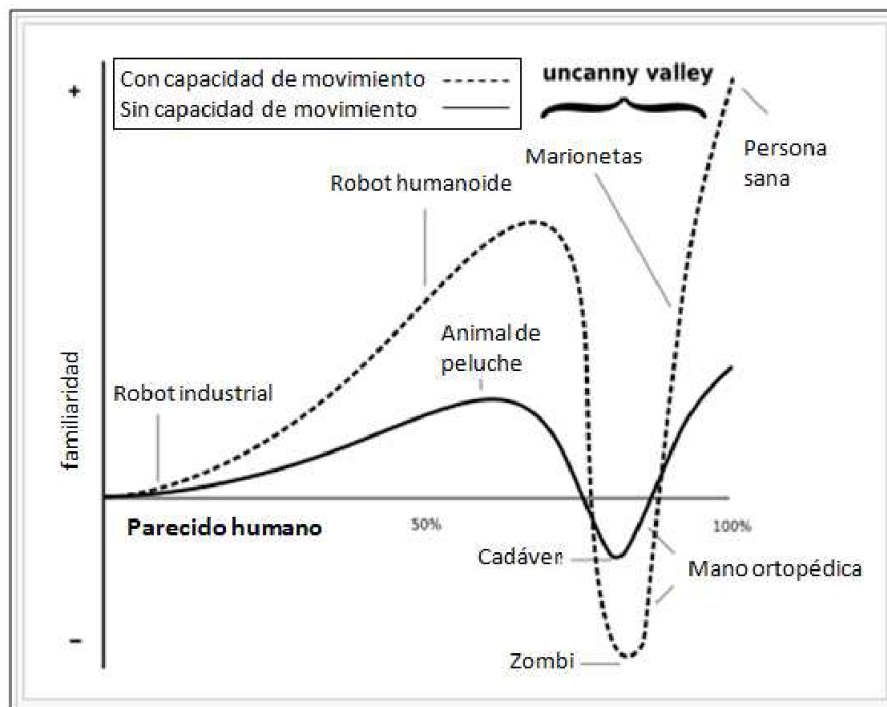


Fig. 2.6. Versión simplificada del gráfico del Uncanny Valley que aparece en el artículo de Energy original de Mori (1970).

En esta gráfica se muestra la respuesta emocional de un ser humano ante el antropomorfismo de un robot. Como podemos ver, existe un pico el cual se refiere a la sensación de familiaridad con el robot, y un segundo pico que se refiere al parecido con el ser humano. Otro aspecto importante a tener en cuenta en la gráfica es la capacidad de movi-

miento, la cual es un signo de vida. Al añadir el movimiento, la forma de la función toma valores mucho mayores a la vez que el mínimo valor decrece.

En el caso de los robots industriales, la característica del movimiento no tiene ningún impacto ya que su función es meramente mecánica, y la capacidad de movimiento muestra si el robot funciona o no. Por otro lado, para los robots humanoides, si sus movimientos son parecidos al de los humanos, su nivel de aceptación crecerá considerablemente.

Finalmente, según Mori, la conclusión de este trabajo es que, al realizar un estudio del diseño de un robot social, se debe considerar como meta el primer pico, ya que, de esta manera se asegura que su aceptabilidad no caiga en el *Uncanny Valley*, como puede pasar con el segundo pico, que, a pesar de tener un grado mayor de aceptabilidad, tiene un riesgo mayor.

El trabajo elaborado por Masahiro Mori ha sido objeto de una gran cantidad de estudios en el mundo de la robótica. Investigadores de la Universidad de California, en San Diego, decidieron comprobar la hipótesis de Mori investigando como reacciona el cerebro de personas cuando observan a robots [33]. El estudio consistía en mostrar imágenes de humanos, robots humanoides y robots sin forma humana a 36 personas elegidas aleatoriamente. Los resultados de la actividad cerebral se muestran en la figura 2.7.

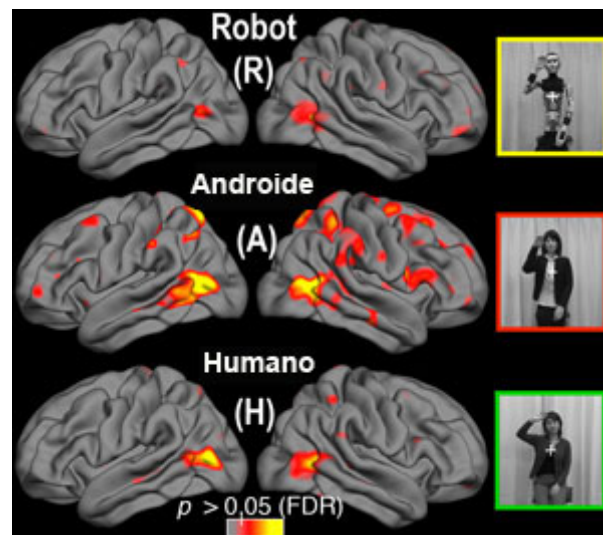


Fig. 2.7. Resultados de la investigación realizada por la Universidad de California, en San Diego [33]. Respuesta cerebral ante interacción: R) Robots, A) Humanoide, H) Humano

Los resultados muestran que, en efecto, el cerebro reacciona de forma diferente ante el humanoide que ante el resto de opciones, esto se debe a que el cerebro reacciona con

una respuesta lógica de la mente humana ante una imagen que no se corresponde con su conocimiento, por lo que los investigadores comprobaron la existencia del " *Uncanny Valley*".

Dentro del robot social, muchos investigadores han concluido que para conseguir la meta de aceptabilidad que se busca, es importante la cabeza del robot, ya que es el principal punto de interacción humano-robot.

Tal y como se expone en *All robots are not created equal* [34], tras un estudio de los rasgos faciales en robots, se pudo comprobar que la presencia de diferentes características faciales aumentaba la percepción de sociabilidad un 62 %. Aquellas características que tuvieron un efecto estadísticamente más significativo en la percepción de la humanidad, fueron la presencia de la nariz, párpados y la boca.

Por lo que se refiere a la cara, debe darse menos espacio a la frente, el pelo, la mandíbula o barbilla, dejando la línea que se sitúa entre la frente hasta la boca domine la cara. Un elemento muy importante a la hora de proyectar humanidad son los ojos, estos deben de incluir cierto grado de complejidad en el detalle de la superficie, la forma del ojo, el iris y la pupila. Algunos estudios como el realizado por P.Ekman [35] también muestran que la presencia de la nariz, boca y párpados contribuyen en gran medida a la percepción de sociabilidad.

Dado el interés de la cabeza del robot, también algunos estudios se han centrado en ella para realizar comparaciones entre los distintos robots sociales. MacDorman e Ishiguro [36] recogen este estudio con la gráfica que aparece en la figura 2.8. Esta gráfica muestra la evolución de la cabeza del robot desde la cara menos humana -que puede ser la del robot Qrio [37]-, hasta una cara con un aspecto humano -como la del robot androide Philip K. Dick [38]-. En la gráfica se puede observar como muestra una correlación con el efecto *Uncanny Valley*, ya que llegado un punto aparece el valle, el cual se debe evitar a la hora de realizar el diseño.

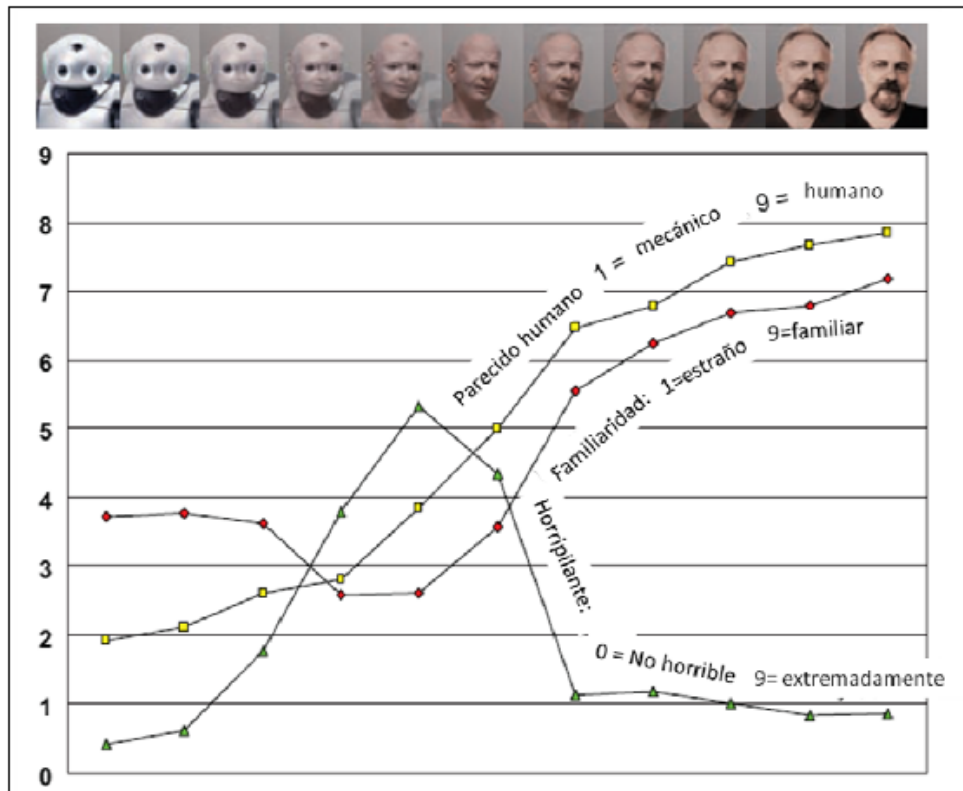


Fig. 2.8. Evolución de la fisonomía desde el robot Qrio hasta el androide Philip K. Dick, creado a la semejanza del propio Philip K. Dick . Fuente : [36].

2.3. Simulación de emociones en robots sociales

Para que un robot pueda llevar a cabo una interacción correcta necesita cumplir con una serie de requisitos a parte de tener un aspecto que resulte amigable. Los robots necesitan la variación de sus expresiones faciales entre otras muchas cosas, como ser capaces de captar y analizar los movimientos del humano.

En el estudio de Russell [39] sobre la apariencia física de un robot se consideran ocho componentes principales: amabilidad, agresividad, timidez, autoritarismo, felicidad, tristeza, peligro y miedo. Estos componentes se pueden dividir según dos factores:

- Intención de comportamiento: este factor considera la amabilidad, agresividad o timidez.
- Factor que estudia las emociones: como felicidad o tristeza.

Para incrementar la percepción sobre cual es el estado actual del robot, es necesario proporcionar una interfaz que cambie dependiendo del estado interno del robot, esto se

consigue gracias a intentar imitar en el rostro las emociones mencionadas anteriormente. Esto se debe a que los humanos tendemos a asociar un comportamiento emocional con determinados cambios en la voz o en la cara del hablante, como podría ser sonreír o gruñir [40].

En relación a lo comentado, Ekman [41], presentó seis expresiones faciales universales asociadas a las emociones básicas: sorpresa, alegría, asco, tristeza, ira y miedo. Ver figura 2.9.

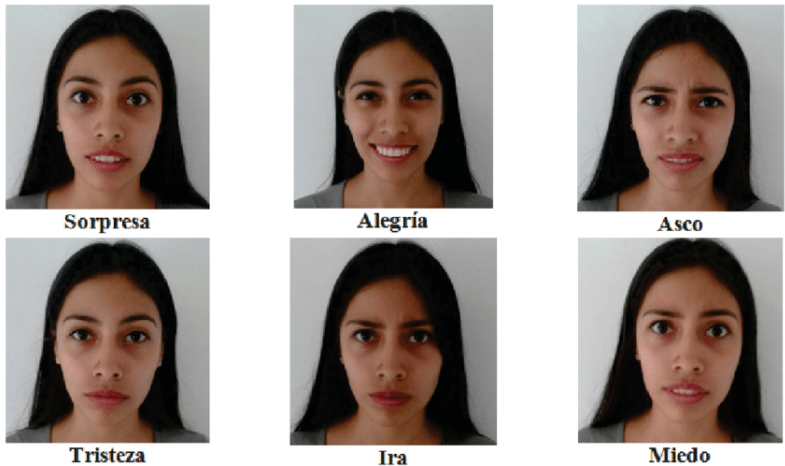


Fig. 2.9. Expresiones faciales universales propuestas por Ekman (1972). Fuente: [42].

Estas expresiones tiene un lenguaje corporal asociado, por ejemplo, la “alegría” tiene asociada diferentes características: boca que puede estar abierta o no, las mejillas pueden estar elevadas, y aparición de arrugas debajo del párpado inferior.

En relación a estas características, Argyle [43] propone que dentro de la expresión facial, las cejas proporcionan una interpretación continua, que es la que se muestra en la tabla 2.1.

Posición de las cejas	Interpretación
Elevadas	Sorpresa o susto
Normales	Sin comentarios
Medio fruncidas	Enfado
Totalmente fruncidas	Tristeza

TABLA 2.1. INTERPRETACIÓN EN FUNCIÓN DE LA POSICIÓN DE LAS CEJAS

Si la cara del robot es capaz de hacer movimientos, la interacción y los sentimientos percibidos por el receptor serán mucho mejores. Nuestro estudio se ha centrado en el desarrollo de un sistema de ojos, boca y cejas capaces de mostrar diferentes emociones y gestos cotidianos, como puede ser parpadear, más adelante se explica con más detalle.

En resumen, dentro de la interacción social el principal componente a la hora de diseñar un robot es su cabeza, ya que gran parte de su interacción con los humanos es a través de ella, gracias a la comunicación visual o verbal. Las principales características del resto del cuerpo dependen de las diferentes culturas, pero éste debe expresar confianza, seguridad y familiaridad con las personas, de manera que éstas no se sientan desprotegidas o en peligro.

La propuesta de diseño del robot presentado en este trabajo fin de grado sigue las especificaciones de Mori [14]. En este caso, para el diseño de la cabeza se ha elegido un tamaño que no sigue las proporciones humanas, una séptima parte del cuerpo, sino que tiene un tamaño superior y es más ancha que alta, para hacerlo más amigable. Además, centrándonos en los rasgos faciales, se descartó la aparición de nariz. En cuanto al tamaño y posición de los ojos, estos siguen el esquema que se expone en *All robots are not created equal* [34], dejando el espacio de los ojos ligeramente más ancho que el radio del ojo. como consecuencia se obtiene una imagen más afectuosa y crea con el usuario un vínculo más cercano.

Siguiendo con el rostro, con la finalidad de conseguir expresiones simples y minimalistas, se ha construido la cara a partir de una caricatura plana, sin cabello, ni nariz, ni orejas, ni las expresiones en la frente y pómulos. De este modo, se consigue un robot con capacidad de interactuar con los humanos sin crear en ellos agresividad o desconfianza.

El rostro abarca toda la pantalla del dispositivo, donde se muestran los distintos tipos de expresiones. Para ello, siguiendo lo anteriormente mencionado en el punto 2.2, se utilizan tres elementos del rostro: ojos, boca y cejas. Por último, las expresiones que se han considerado, en relación con el estudio de P. Ekman [41], son: alegre, normal, triste, dormido y asustado. Las cuales se muestran en la figura 2.10.

Estas figuras han sido creadas gracias al programa Inkscape, el cual se explica más adelante y una fuente de imágenes del programa Pinterest. Partiendo de las imágenes encontradas en la fuente de Pinterest se han retocado las imágenes para que cumplan con los requisitos propuestos.



Fig. 2.10. Diferentes expresiones planteadas para el rostro del robot.

3. DISEÑO CONCEPTUAL DEL SISTEMA

La construcción de la cabeza del robot se ha basado en un diseño con aspecto amigable y que provoque el acercamiento a los seres humanos de una manera natural. Además, se ha tenido en cuenta el tamaño y diseño del resto del cuerpo para su construcción, ya que debía de mantener una concordancia con el tronco, para poder ajustarse a sus medidas y condiciones de movimiento.

El diseño de la cabeza se divide en: el diseño del rostro del robot (sección 3.1) y en el diseño de la carcasa de la cabeza con el programa FreeCad y su impresión 3D (sección 3.2).

3.1. Diseño del rostro del robot

Para el diseño de la cara, se ha optado por una cara iconográfica que únicamente consiste en un par de ojos con dos cejas y una boca, ya que, de ese modo, es universalmente reconocible. Además, con esos elementos se pueden representar una amplia gama de emociones simples, que son muy útiles en la interacción con humanos (figura 2.10).

Como se ha comentado en la sección 2.3, la cara consta de los rasgos faciales básicos. Además, hemos añadido párpados, que nos permiten programar un abanico todavía más amplio de expresiones, como pueden ser guiñar un ojo o cerrar ambos para simular el parpadeo.

La pantalla donde se muestran las diferentes caras está conectada al robot para que según lo que haga éste muestre una cara u otra. El comportamiento de esta pantalla, donde se mostrarán las expresiones faciales, depende de la velocidad que lleve el robot además, de si ha conseguido llegar con éxito al destino que se propone o no.

En el siguiente flujo-grama de la figura 3.1 se muestran las expresiones faciales dependiendo de los cambios emocionales.

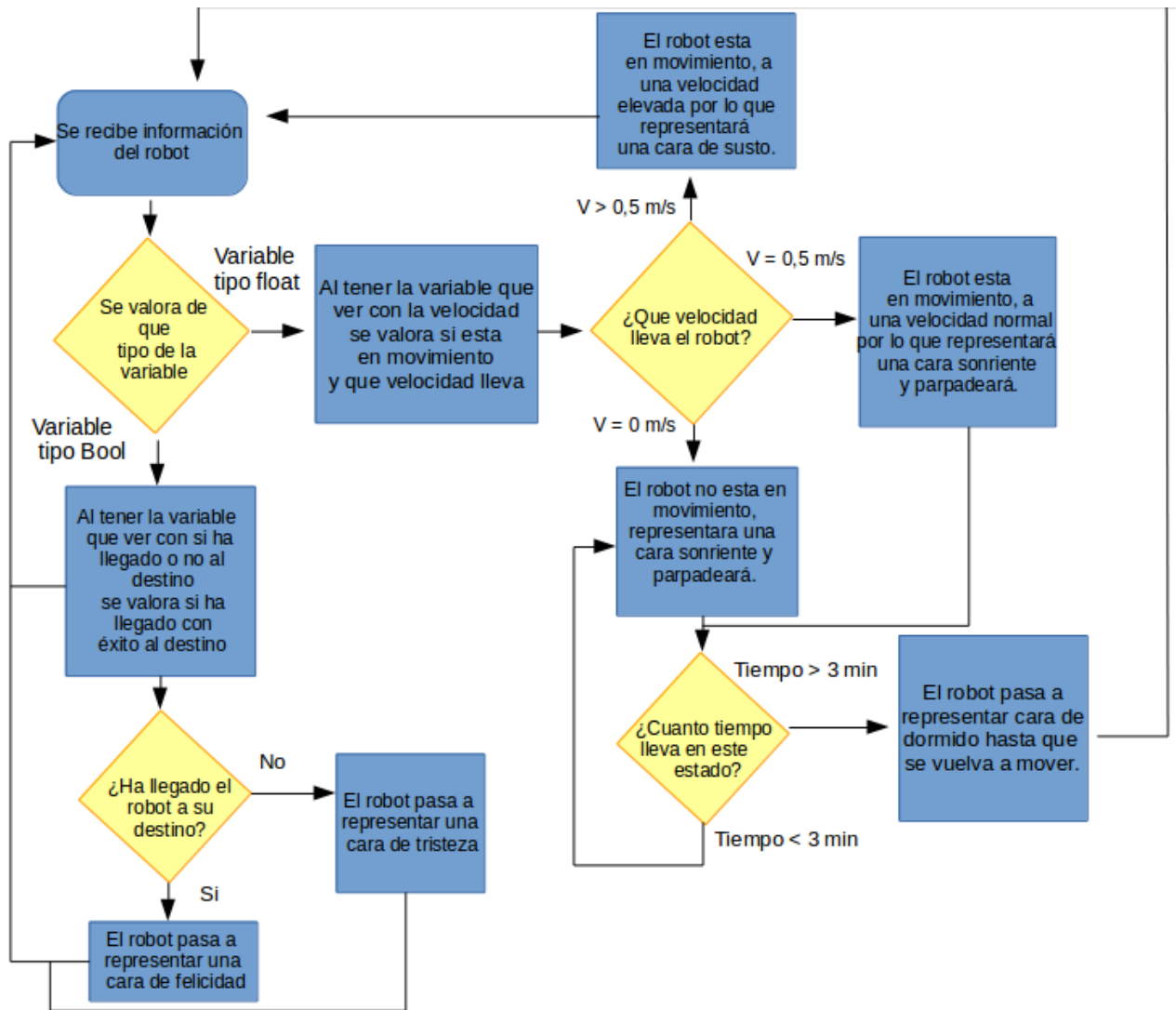


Fig. 3.1. Diagrama de flujo explicando de manera conceptual el programa de la representación de la cara del robot.

En primer lugar, el sistema de cara recibe información del estado del robot. En el caso de que se reciba una información booleana se relaciona con la llegada con éxito o fracaso al destino deseado por el usuario, mostrando en ambos casos una cara según su estado. Por otro lado, si la información recibida es de tipo Twist, nos indica la velocidad que lleva el robot. Según esta velocidad existen tres estados diferentes mostrando distintas expresiones. Por ultimo, según el tiempo que lleva el robot parado aparece un nuevo estado de dormido.

3.2. Diseño de la carcasa de la cabeza

Para la carcasa de la cabeza se han diseñado dos piezas por separado. En primer lugar, la cabeza que se encarga de dar soporte a los diferentes componentes que llevan a cabo la interacción del robot. Y por otro lado, la base que lleva a cabo la unión de la cabeza con el cuerpo del robot.

Para el diseño de la carcasa de la cabeza, se han utilizado los trabajos anteriormente comentados para guiarnos con las proporciones y formas ideales del cuerpo humano; siendo la cabeza del robot más grande que las proporciones ideales del cuerpo humano. Estas proporciones nos dicen que la cabeza debe medir aproximadamente una séptima parte de la altura total del individuo, por lo tanto, teniendo en cuenta lo anterior, y que nuestro robot mide 150 cm de altura, el resultado es que la cabeza debe medir aproximadamente unos 21 cm como máximo.

Además, es importante tener en cuenta los grados de inclinación de la cabeza y la forma que tiene según las dimensiones de la pantalla y la altura del robot.

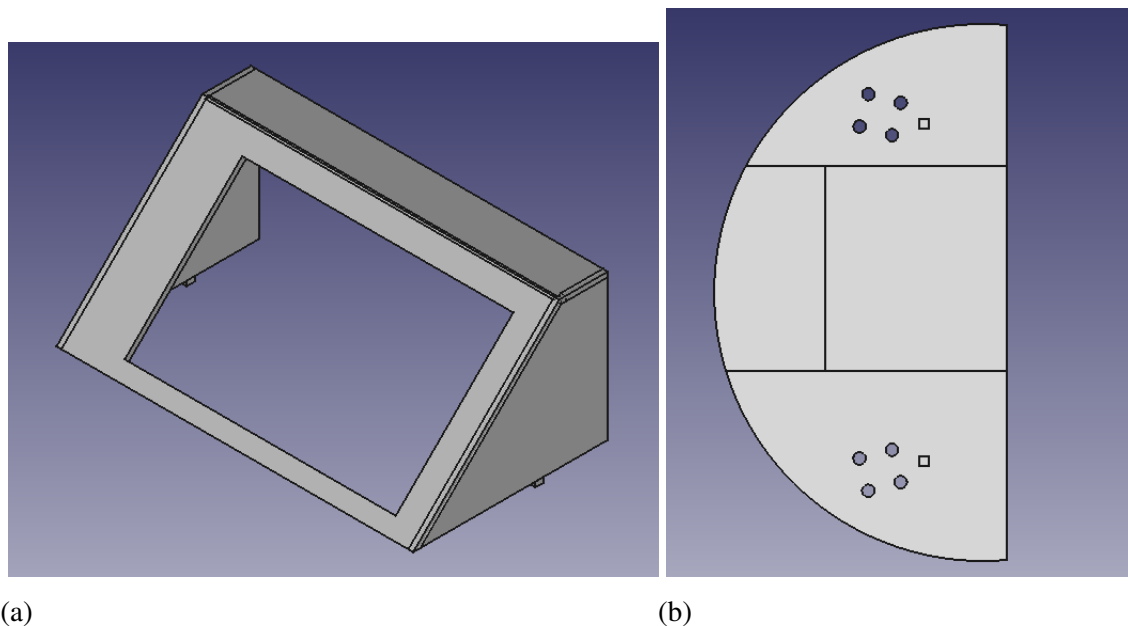


Fig. 3.2. a) Diseño de la carcasa de la cabeza mediante el programa FreeCad. b) Diseño en FreeCad de la base que une la cabeza con el cuerpo del robot.

Como se muestra en la figura 3.2 a), la cabeza tiene un ángulo de inclinación de 45° , ya que al tener una altura de 150 cm es necesario este ángulo para que las imágenes de

la pantalla se puedan ver con claridad. Además, tiene una forma rectangular, debido al tamaño y la forma de la pantalla LCD.

Por otro lado, la base sigue las medidas del cuerpo del robot para poder adaptarse a su forma que pueda estar sujeto mediante taladros. Por último, en la figura 3.2 b), la base de la cabeza se ha hecho siguiendo las coordenadas y medidas del cuerpo del robot.

Una vez que hemos visto a nivel general cual va a ser el funcionamiento y el diseño principal del robot, vamos a pasar a ver en detalle el desarrollo de la expresión visual y el diseño de la carcasa de la cabeza en las siguientes secciones.

4. DESARROLLO DE LA EXPRESIÓN VISUAL DEL ROSTRO DEL ROBOT

En este capítulo se explica la implementación de la interfaz gráfica de la pantalla. Esta implementación se realizó con el objetivo de lograr que las representaciones faciales tengan coherencia y versatilidad expresiva. De este modo, el robot puede expresar una amplia variedad de emociones, dando así, la posibilidad de cambiar de expresión facial según sus cambios emocionales.

La implementación de la pantalla se realiza a través de Arduino. Se ha utilizado ROS para conectarse con el robot, ya que éste está totalmente implementado en ROS como se explica en la sección 4.3.

En la construcción del programa, las imágenes se dividieron en 3 partes: por un lado, los ojos, por otro lado, las cejas y, por último, la boca. Cada una de estas partes tiene un tamaño diferente según los píxeles de la pantalla, tal y como se explica en la sección 4.4 donde también se puede ver de qué manera se ha implementado la transición de imágenes. Se toma como referencia la expresión de cara "normal" para que después pase a las diferentes expresiones que se proponen en la figura 4.13.

Además de todo lo mencionado, en este capítulo se explica los elementos para el desarrollo electrónico y el *software* necesario para la implementación del programa.

4.1. Ingeniería de requerimientos para el desarrollo electrónico

El sistema electrónico que lleva a cabo el control de expresiones del robot descrito se compone de:

- Un Arduino Mega sobre el que estará el programa de control de expresiones en lenguaje Arduino. El Arduino va alimentado mediante una fuente de alimentación a una tensión de 5 V.
- Una pantalla TFT LCD compatible con el *hardware* libre de Arduino.

- Una tarjeta de expansión que realiza una función de escudo entre el Arduino y la pantalla, ya que mediante los pines de salida PWM y los pines digitales del Arduino se interconecta con la pantalla LCD.
- Una tarjeta SD conectada directamente a la pantalla, esta tarjeta sirve como fuente de imágenes para el programa.

El interconexionado de los diferentes elementos puede observarse en la figura 4.1

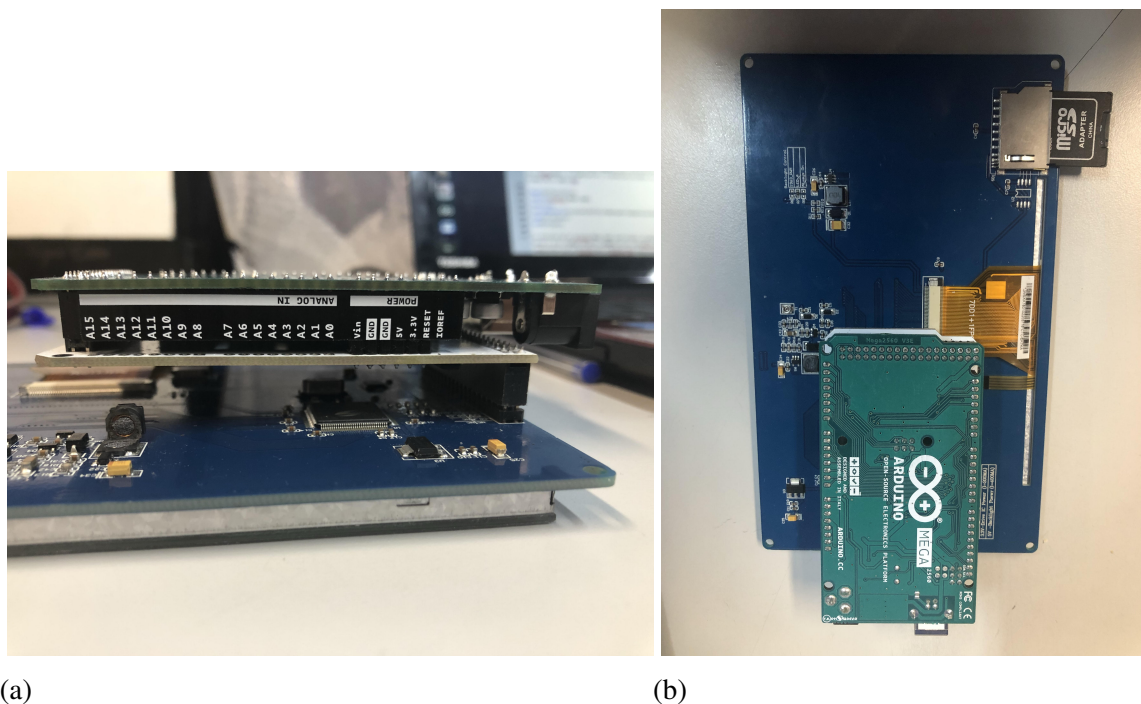


Fig. 4.1. Interconexionado del sistema electrónico: a) Vista lateral donde se muestra la adaptación de la pantalla y el Arduino mediante la tarjeta de expansión. b) Vista inferior donde se puede ver la conexión de la tarjeta SD.

4.1.1. Arduino

El Arduino que vamos a utilizar es el modelo Mega 2560, como el que aparece en la figura 4.2, y está compuesto por un microprocesador AVR de 8 bits ATmega2560. La alimentación y la transmisión de datos de la placa se hace a través de un cable USB.

La elección del Arduino Mega como controlador se decidió en base a su bajo coste y a su código abierto, en comparación con el Arduino Uno o Arduino Due tiene una memoria

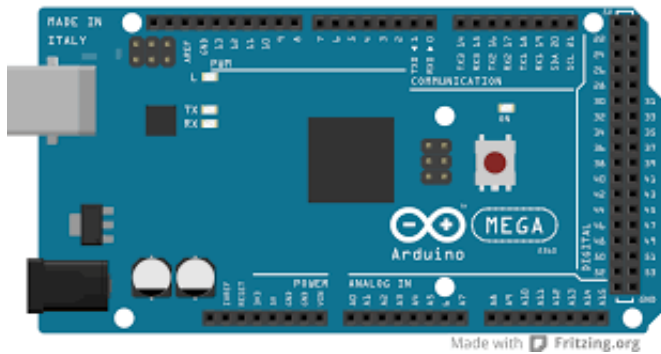


Fig. 4.2. Arduino MEGA 2560

mayor tanto memoria SRAM (8 Kb) como FLASH (256 Kb) por lo que para mostrar imágenes era la mejor opción.

4.1.2. Módulo de Pantalla Táctil TFT LCD 800x480

La pantalla que se utiliza para mostrar las imágenes es el módulo TFT LCD SSD1963 de 7.0 pulgadas que podemos ver en la figura 4.3. Esta pantalla proporciona una imagen colorida de alta calidad y ángulo de visión amplio. La pantalla tiene como principales características técnicas:

- LCD Tipo: TFT Transmisivo Normal Blanco
- Tamaño del módulo: 186 * 106 * 23 mm / 7.32 * 4.17 * 0.9 "
- Tamaño de la pantalla: 154 * 86 mm / 6.06 * 3.38 "

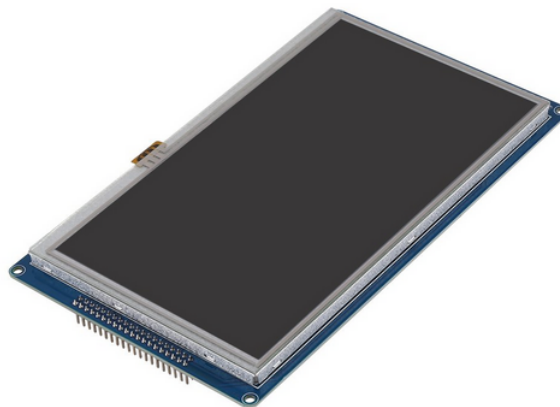


Fig. 4.3. Pantalla TFT LCD 800x480

Como se indica en el manual de la pantalla SSD1963 [44] (añadido al ANEXO B), es necesario que esté el pin del LED_A siempre conectado a corriente para que la pantalla

se encienda, con lo cual debe también estar conectado el pin a tierra, el pin de 3,3V y el pin de 5V. El esquema de este conexionado se muestra en la sección 4.1.3.

4.1.3. Tarjeta de expansión Escudo LCD táctil para Arduino Mega

Para conectar el Arduino y la pantalla LCD utilizamos la tarjeta de expansión de la figura 4.4, que es un escudo para Arduino Mega. Gracias a este módulo se puede conectar directamente el TFT01 en el escudo y colocarse en la placa Arduino MEGA como se muestra en la figura 4.5.

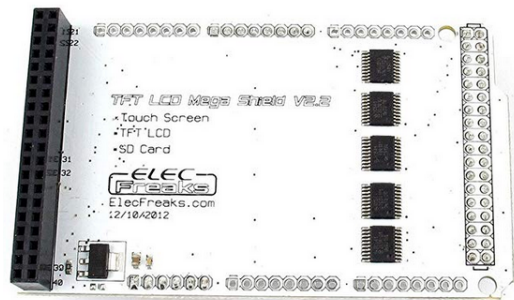


Fig. 4.4. Tarjeta de expansión Escudo LCD

Esta tarjeta de expansión es necesaria debido a que el pinedo de la pantalla TFT LCD no coincide ni con el número de pines digitales ni con la posición de estos en el Arduino Mega 2560, además de adaptar la tensión de funcionamiento gracias a diferentes Buff/DVR que lleva integrados (ver ANEXO A).

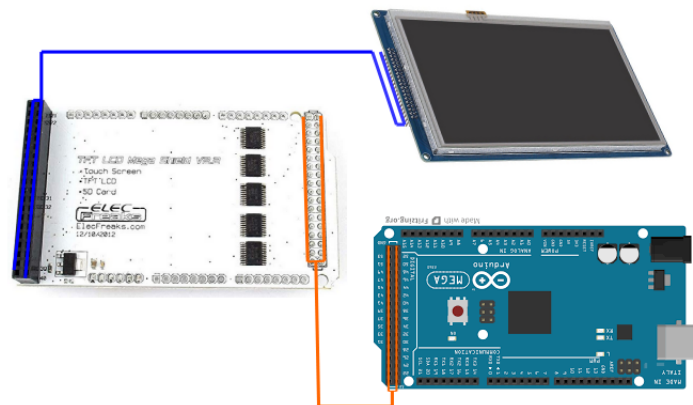


Fig. 4.5. Esquema de como conectar la pantalla LCD con el escudo para su correcto funcionamiento

4.1.4. Tarjeta de Memoria SD de 2 GB

Utilizaremos la tarjeta Transcend TS2GSDC con capacidad de 2 GB para almacenar todas las imágenes.



Fig. 4.6. Tarjeta de Memoria SD Transcend de 2 GB.

En un principio se optó por no usar tarjeta SD ya que existe la opción de convertir las imágenes en arrays que puedan ser entendidos por nuestro Arduino. Estos arrays tienen una variable por cada píxel que tenga la imagen. Los archivos creados en este proceso vienen preparados para cargar el array en la memoria flash del Arduino. Sin embargo, las imágenes que se pueden utilizar con esta opción deben de ser bastante más pequeñas que el tamaño que necesitamos para nuestra pantalla.

4.2. Ingeniería de requerimientos para el desarrollo *software*

En este apartado se explica el *software* previo necesario para poder desarrollar las funcionalidades del robot.

4.2.1. Robot Operating System

ROS [45], es un sistema operativo (o framework) de código abierto para el desarrollo de *software* robótico. Su propósito es facilitar el desarrollo de *software* para robots a distintos niveles y así conseguir que el mismo código pueda ser utilizado desde diferentes ordenadores. Esta característica es una de las razones por las que utilizamos ROS, ya que permite que los diferentes sistemas tengan diferentes nodos (pudiendo tener diferente arquitectura), lo que a su vez nos da flexibilidad a la hora de trabajar. La versión de ROS utilizada para este proyecto es la Kinetic Kame, que fue lanzada el 23 de mayo de 2016. Trabajamos en esta versión ya que está dirigida a la distribución Ubuntu 16.04 (con

arquitectura armv7l, que es la que estamos utilizando.

De manera general, y como hemos mencionado anteriormente, el funcionamiento de ROS se basa en la intercomunicación de distintos procesos que pueden estar localizados en distintos ordenadores conectados a una red con tipología *peer-to-peer*. Un ejemplo de estructura P2P es la mostrada en la figura 4.7.

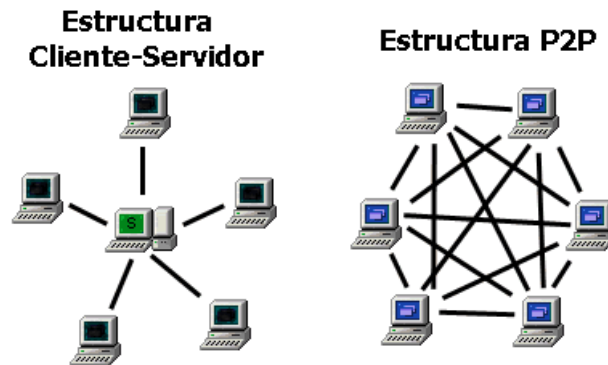


Fig. 4.7. Ejemplo de red P2P [46].

■ Nodos

Los nodos son archivos ejecutables con distintas funcionalidades dentro de un paquete. Estos archivos pueden comunicarse entre sí gracias a la capacidad de publicación y suscripción a un *topic*. De este modo, se consigue realizar una tarea específica dentro del conjunto que conforma la red.

Los nodos se comunican intercambiando estructuras de datos llamadas mensajes. Dichos mensajes pueden estar compuestos por los tipos de datos estándar (integer, boolean, character, etc.) o por diversas estructuras de datos, compuestas por estos mismos tipos.

En nuestro trabajo, aparte de utilizar un tipo de mensaje compuesto por un dato boolean, para la llegada con éxito al destino, utilizamos un mensaje de ROS compuesto de varias estructuras para identificar la velocidad del robot. Se trata del mensaje `geometry_msgs/Twist.msg`. y su estructura interna se muestra en la figura 4.8.

■ topics

Los *topics* son buses por los cuales los nodos intercambian mensajes. Usando *Subscriber*, un nodo puede acceder a la información del *topic*, mientras que usando *Publisher* se puede mandar o publicar información en un *topic*.

```
geometry_msgs/Vector3 linear
float64 x
float64 y
float64 z
geometry_msgs/Vector3 angular
float64 x
float64 y
float64 z
```

Fig. 4.8. Estructura de datos *geometry_msgs/Twist.msg*.

Además de este tipo de comunicación, en el caso en el que un nodo necesite una respuesta a una solicitud, la comunicación se hace por medio de los servicios. Un servicio funciona con el intercambio de un mensaje compuesto por una solicitud y su correspondiente respuesta. El nodo interesado en recibir la respuesta (cliente) envía ese mensaje a un nodo que está ofreciendo el servicio bajo un nombre específico. Cuando este último recibe dicho mensaje, procede a procesar la información recibida y envía los resultados al cliente. El ejemplo más sencillo de servicio es una suma, donde el cliente llama al servicio enviándole dos números y el nodo a cargo del servicio se encarga de sumarlos y devolver el resultado.

En la figura 4.9 se puede ver un esquema simplificado del funcionamiento de *topics* y servicios.

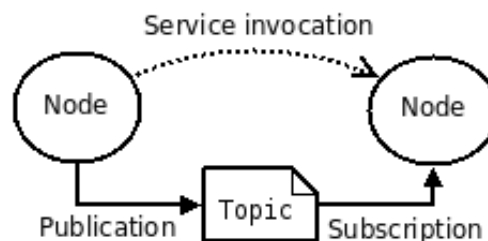


Fig. 4.9. Ejemplo comunicación ROS. Tanto usando *topics* (flujo inferior) como servicios (flujo superior)

4.2.2. Arduino

Arduino es un *software* de código abierto destinado a la programación de sistemas embebidos, además dispone de un gran abanico de librerías para el control de sensores y actuadores.

Dado que se necesitaba la comunicación con ROS, se instaló “roserial” (en su versión 0.7.6), ya que este protocolo contiene el paquete roserial_arduino, en el cual se encuentran las extensiones específicas para poder ejecutar roserial_client en un Arduino. Usando este paquete, podemos utilizar ROS directamente con el IDE de Arduino, pudiendo publicar y suscribirse directamente a los mensajes ROS.

Antes de mostrar el proceso de comunicación entre Arduino y ROS, es necesario permitir esta comunicación copiando en el entorno de compilación de Arduino la biblioteca “ros_lib”. Esto se hizo del siguiente modo:

En el interfaz de Arduino nos situamos en la ventana “Programa”, después clicamos en “Incluir Librería” y, por último, “Añadir Librería”. Buscamos roserial y una vez instalado debería aparecernos como en la figura 4.10.

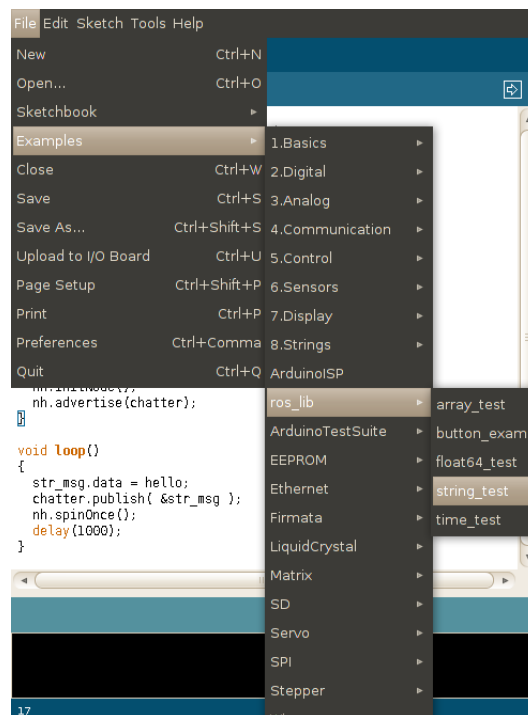


Fig. 4.10. Pantalla de herramientas del programa Arduino

4.2.3. Inkscape

Inkscape es un editor de gráficos libre y de código abierto. Con este programa se han realizado las figuras de las diferentes expresiones, ya que, mediante Inkscape se puede crear y editar gráficos, logotipos, e ilustraciones complejas.

Partiendo de la fuente de imágenes proporcionada en Pinterest, se ha conseguido modificar las características que se han visto oportunas tras el estudio de los rasgos faciales, importantes para una interacción humano-robot adecuada. Además, gracias a este editor de gráficos se ha conseguido separar cada rasgo facial con los píxeles que nos convenía en cada caso.

4.3. Integración ROS/ARDUINO

Una vez explicados los elementos principales que se van a integrar en este proyecto voy a explicar el proceso de integración del programa principal con el resto del robot.

El sistema de expresiones de la cara implementado en Arduino necesita conocer en todo momento la cinemática de la plataforma móvil sobre la que va montado, es decir, su velocidad, la cual, está implementada principalmente en ROS. De este modo, el sistema de Arduino se sirve del framework ROS para recibir esta información y actuar en consecuencia.

Para esta integración descrita se hace uso de ROS. Se establecen ambas partes como Nodos que se envían mensajes entre ellos, como se indica en el siguiente diagrama de la figura 4.11.

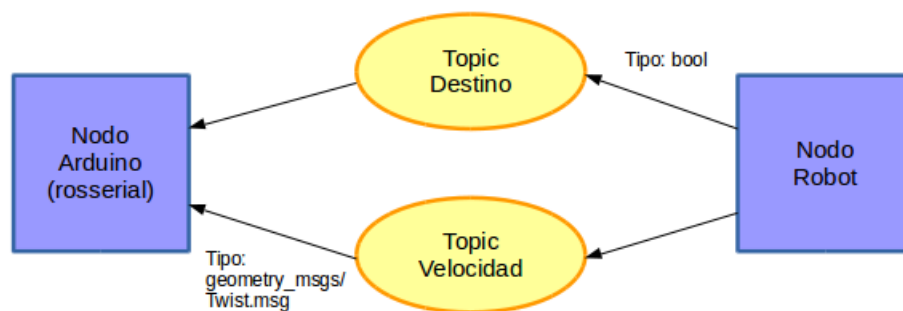


Fig. 4.11. Diagrama explicativo de la integración ROS/Arduino.

Se han creado dos nodos emisores de prueba mediante los cuales se publica en ROS, por un lado, si el robot ha llegado a su destino en forma de *Bool*, y, por el otro, las componentes de las velocidades lineales y angulares de la plataforma móvil en forma de variable de tipo *Twist* bajo el *topic* Velocidad. Este nodo se va a comportar como publisher enviando información para que el nodo Arduino suscrito a los topics mencionados lo

reciba. Esto se consigue gracias al paquete `rosserial_arduino`. Este paquete es un protocolo de comunicación ROS que permite usar ROS directamente con el IDE de Arduino. De este modo, se permite que nuestro Arduino sea un Nodo ROS completo y así poder publicar y suscribirse directamente a mensajes ROS. Para la instalación de este paquete es necesario ejecutar en el terminal una serie de sentencias y crear la librería `ros_lib` para copiarla en el entorno de compilación de Arduino.

Tras haber creado el nodo Arduino suscrito a los *topics* con la información necesaria, el programa principal de este trabajo va revisando en cada bucle del loop principal la recepción de nuevos mensajes para el destino o la velocidad y, en caso de recibirlos, se llama a unos subprogramas conocidos como *callbacks*. Estas funciones se explican en la sección 4.5.

4.4. Expresiones implementadas

Como ya se ha descrito anteriormente, el rostro del robot se dividió en 3 partes: ojos, cejas y boca. Estas divisiones del rostro del robot se hacen según los píxeles totales de la pantalla (800x480 píxeles), siendo estas imágenes por separado y las 3 partes juntas de un tamaño acorde a la pantalla (ver figura 4.12).

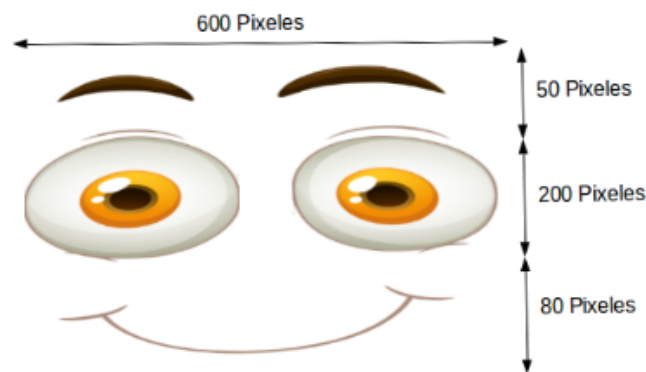


Fig. 4.12. División en píxeles de las diferentes partes que constituyen la cara del robot.

Para el cambio de emociones, se consideró la expresión sonriente como el estado neutral, de manera que todas las transiciones entre las imágenes pasan por la expresión neutral.

Para realizar cambios de una expresión facial a otra, cuando el robot experimenta un cambio en sus emociones, se realiza un refresco de la parte del rostro que se quiere

cambiar por separado (es decir, se refrescan por separado los ojos, boca y cejas). De esta manera, se evitan cambios bruscos generando cambios progresivos entre las diferentes expresiones. Todo ello a partir de la expresión neutral.

Adicionalmente, se realizaron diferentes ajustes, en cuanto a tiempos de duración de las transiciones, hasta lograr que dichas transiciones se observen de forma natural como se explica en la sección 4.5.3.

Desde la expresión neutral se pueden realizar 4 transiciones: dormido, asustado, feliz o triste. Como podemos ver en la figura 4.13, existen dos estados adicionales a las 5 emociones que puede expresar el robot, estos estados son: el estado de parpadeo y el estado de muy asustado.

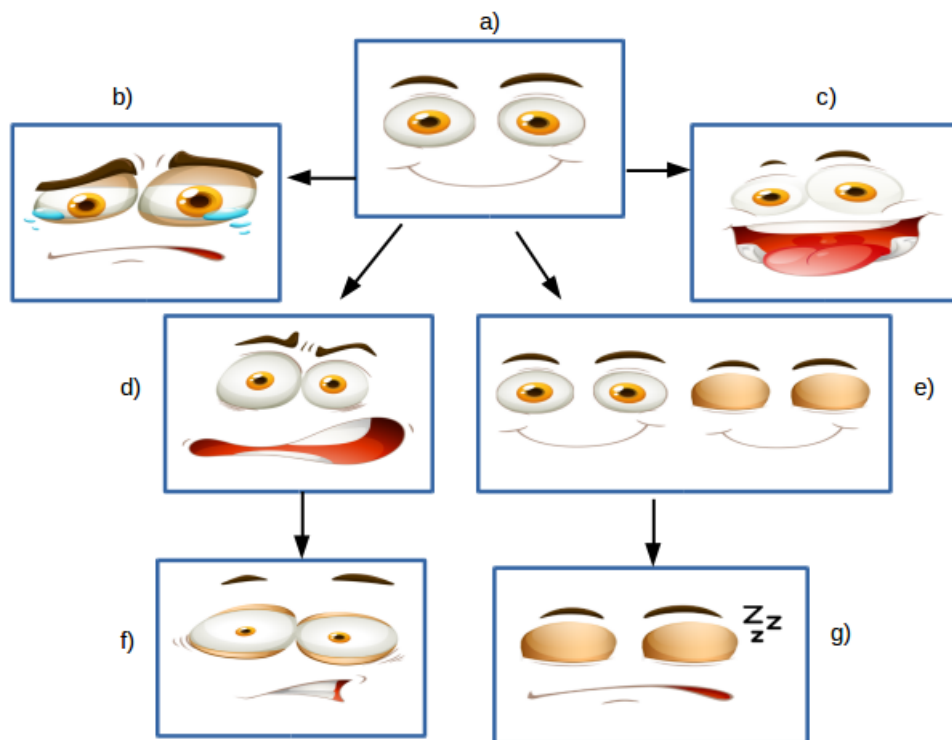


Fig. 4.13. Transición de las diferentes expresiones del robot. a) Neutral b) Triste c) Feliz d) Asustado e) Parpadeo f) Muy asustado g) Dormido.

- **Estado neutral:** como se ha mencionado anteriormente, este estado es el inicial y muestra una cara sonriente con los ojos abiertos. En este estado se está a la espera de que reciba alguna variable para cambiar de expresión o pasar al estado de parpadeo, si no recibe ninguna información.
- **Estado de tristeza:** para este estado se muestra una cara de tristeza simulando el

lloro, ya que se da a entender que el robot no ha llegado a su destino. En esta expresión se realiza una transición de imágenes por separado de la boca, ojos y lágrimas, dejando en la misma imagen las cejas y los ojos y eliminando las cejas de la cara sonriente. De este modo, se crea una sensación de pena más real.

- **Estado de felicidad:** en el estado de felicidad se da a entender que el robot ha llegado a su destino y, para ello, se realiza una transición de las imágenes de boca, cejas y ojos por separado.
- **Estado de parpadeo:** a la hora de realizar las expresiones y la transición de éstas se ha buscado naturalidad y cercanía con la persona con la que esta interactuando. Para ello, se ha creado un estado de parpadeo, el cual, cuando el robot está parado o a una velocidad lenta se produce una secuencia de imágenes imitando el parpadeo de un ser humano, figura 4.13 e). En esta expresión solo se ha cambiado la imagen de los ojos, de este modo se consigue naturalidad y una mayor rapidez en el refresco de la imagen completa de la cara.
- **Estado dormido:** el estado de dormido surge cuando el robot lleva un tiempo determinado parado. Esta expresión se consigue cambiando únicamente la imagen de la boca y añadiendo el símbolo de dormido (Z Z Z) por separado ya que este estado surge del estado de parpadeo que ya tiene los ojos cerrados.
- **Estado asustado:** al ir el robot a una velocidad elevada su estado cambia y se muestra una expresión de asustado, realizando una transición de imágenes en cada una de las partes del rostro por separado.
- **Estado muy asustado:** cuando el robot supera su velocidad máxima, la cara representa una expresión diferente. Esta expresión cumple con los requisitos mencionados en apartados anteriores, como puede ser la elevación de las cejas. En esta expresión, al contrario que en la del estado de parpadeo, se realiza una transición de imágenes del rostro completo.

4.5. Implementación del programa

Dado que esta programación se hace en código de Arduino, veremos de una forma global su funcionamiento. El lenguaje de programación de Arduino está basado en C++

y se puede dividir en tres partes principales: estructura, valores (variables y constantes) y funciones [45].

El programa “.ino” de Arduino tiene una estructura con dos funciones principales: *setup()* y *loop()*. En la función *setup()*, se inician las variables y los *pinMode*, esto ocurre solo una vez cuando enciendes la placa del Arduino. La función *loop()* se ejecuta cíclicamente de manera continua, esto permite que el programa cambie y responda en el tiempo.

Como se ha comentado en la sección 4.3, en este trabajo se pretende la integración del programa implementado en Arduino de la pantalla con el programa del PC Master implementado en ROS del cuerpo del robot. Para ello es necesario incorporar las siguientes librerías:

- La librería de ROS (*ros.h*), gracias a ella conseguimos que todo aquello que se produce en el robot tenga repercusión en el programa de Arduino.
- Las librerías *std_msgs/Bool.h* y *geometry_msgs/Twist.h* para poder trabajar con las variables que constituyen los mensajes recibidos.

Desde el PC Master se crea un nodo de control del robot, el cual se encarga de publicar en el *topic* el destino y velocidad del robot. El nodo Arduino se suscribe a este *topic*, y actúa mostrando las diferentes imágenes según su estado. De este modo, el Arduino funciona como *Subscriber* y el PC Master como *Publisher*.

A continuación, se expone la estructura del programa seguido por la explicación del sistema implementado para el uso de las diferentes expresiones del rostro:

1. Inclusión de librerías: entre las necesarias, se encuentran las librerías para la definición de los datos que van a ser pasados como mensaje entre el *Publisher* y *Subscriber* del trabajo, que como hemos mencionado, son de tipo *Bool* y *geometry_msgs/Twist.msg*.
2. Declaración de las variables globales.
3. *Callbacks* de ROS (explicados en la sección 4.5.1):
 - Recepción del éxito o fracaso de la llegada al destino.
 - Recepción de la velocidad de la plataforma móvil.

4. Inicialización (*setup()*):

- Se inicializan los pines de control de la tarjeta SD y de la pantalla.
- Se incluye la función *Serial.begin(9600)*, que establece la velocidad de datos en bits por segundo (*baud*) para la transmisión de datos en serie.
- Se inicializan las comunicaciones con ROS y se llevan a cabo las suscripciones a los *topics* de destino y velocidad, mediante las siguientes funciones:
 - *InitNode()*: inicia el nodo Arduino.
 - *Subscribe()*: se suscribe a los *topics* que le indiquemos.

5. *Loop* principal:

- *spinOnce()*: procesa nuestras devoluciones de llamada.
- Si es la primera ejecución del *loop*: se muestra por pantalla la expresión neutral (rostro sonriente).
- Si el robot se encuentra parado o a una velocidad lenta: se entra en el estado de parpadeo o dormido, explicado en la sección 4.5.2.

4.5.1. Implementación del Subscriber

Como se ha mencionado, el Arduino funciona como un nodo Subscriber. Su principal función es suscribirse al *topic Destino* y *Velocidad* que indican la acción que se debe ejecutar.

Topic Destino

En este *topic* al llegar un mensaje se produce una llamada a la función “*Callback_Destino*”, que es la encargada de mostrar las diferentes imágenes en la pantalla según la variable recogida. Al llegar un mensaje True, el Arduino muestra la imagen que expresa felicidad, y al llegar un mensaje False expresa tristeza (ver figura4.14).

Topic Velocidad

Los datos recogidos por el *topic Velocidad* son de tipo *geometry_msgs/Twist.msg*. Debido a esto, es necesario elegir el dato que nos interesa para la velocidad. Este tipo de variable consta de dos vectores (lineal y angular), en este trabajo utilizamos el tipo *geometry_msgs/Vector3linear* y la coordenada *linear.x*

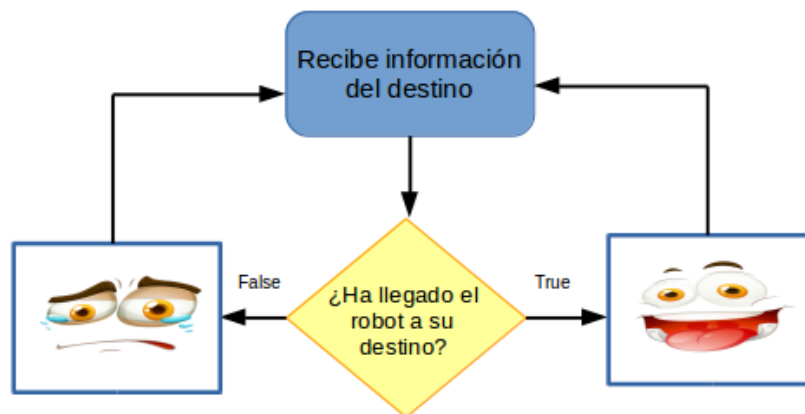


Fig. 4.14. Diagrama del funcionamiento de la función Callback_Destino.

Al igual que en el *topic Destino*, se produce una llamada a la función “Callback_Velocidad”. En esta función, tras valorar la velocidad recibida, se muestra la imagen que corresponde en cada situación (ver figura 4.15). A continuación, se realizan las siguientes sentencias que nos sirven para el estado de parpadeo y dormido:

- Guarda el segundo en el que se capta la velocidad, para valorar el tiempo que pasa el robot parado, como veremos más adelante.
- Se cambia la variable boolean Parado y Dormido a falso, de esta manera sabremos en que estado estamos.

4.5.2. Implementación del estado de parpadeo y dormido

Como se ha mencionado en la sección 4.4, en nuestro trabajo existe el estado de parpadeo que complementa el estado neutral. A este estado se llega una vez pasamos más de 1 minuto en el estado neutral (cuando estamos parados o a una velocidad lenta).

La secuencia de parpadeo se produce 10 veces por minuto, ya que en general, según menciona M.J Doughty [47], entre cada parpadeo se produce un intervalo de 2-10 segundos en personas adultas.

Por último, al estado de dormido solo se llega una vez se cumplan las siguientes condiciones:

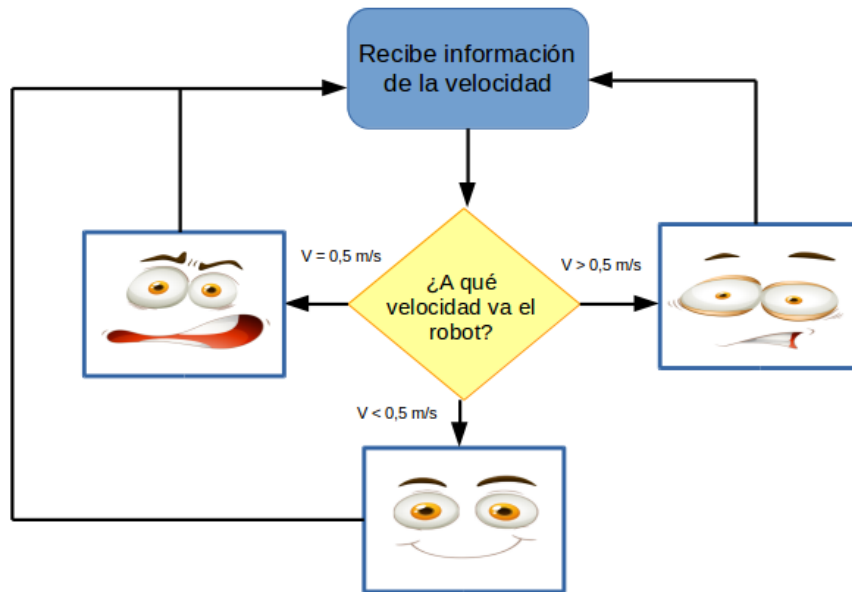


Fig. 4.15. Diagrama del funcionamiento de la función Callback_Velocidad.

- Estar parado (Velocidad = 0 m/s).
- Llevar en el estado de parpadeo más de 3 minutos.

4.5.3. Implementación de las imágenes

La parte principal del programa consiste en mostrar las imágenes por la pantalla, estas imágenes son mostradas gracias a las siguientes librerías:

1. **Librería UTFT_SdRaw**: realiza la función de dibujar por pantalla las diferentes caras que va a tener el robot.
2. **Librería UTFT**: envía las imágenes de la tarjeta SD a la pantalla.

La principales sentencias de estas librerías, que son las utilizadas en este trabajo son:

- **UTFT(Modelo, RS, WR, CS, RST)**: esta sentencia es el constructor de la clase principal cuando se utilizan módulos de visualización. En la tabla 4.1 se muestran los diferentes parámetros que la componen quedando con los pines correspondientes a nuestra pantalla de esta manera :

UTFTmyGLCD(SSD1963_800ALT, 38, 39, 40, 41)

RS	Pin correspondiente a nuestra pantalla encargado de mantener la información que sale por la pantalla, o donde el controlador de ésta busca los datos a mostrar.
WR	El pin de “lectura/escritura” selecciona el modo de lectura o de escritura (en nuestro caso escritura).
CS	Pin correspondiente a nuestra pantalla encargado de seleccionar el chip.
RST	Pin encargado de restablecer la información de la pantalla.

TABLA 4.1. DEFINICIÓN DE LOS DIFERENTES
PARÁMETROS DEL CONSTRUCTOR DE LA LIBRERÍA UTFT.

- **UTFT_SdRaw myFiles(&myGLCD):** Esta sentencia inicia una instancia para poder leer las imágenes que guardaremos en nuestra tarjeta SD.
- **myFiles.load(x, y, sx, sy, Imagen.RAW):** herramienta de dibujo de la librería UTFT_SdRaw. En la tabla 4.2, se muestran las definiciones de los parámetros.

X	Coordenada x de la esquina superior izquierda del mapa de bits
Y	Coordenada y de la esquina superior izquierda del mapa de bits
Sx	Ancho del mapa de bits en píxeles
Sy	Altura del mapa de bits en píxeles

TABLA 4.2. DEFINICIÓN DE LOS DIFERENTES PARÁMETROS
DE LA HERRAMIENTA DE DIBUJO.

Como se ha mencionado en la sección 4.4. la cara del robot se ha dividido en 3 partes a la hora de mostrarlas por pantalla. Este proceso se hace debido a que las imágenes se transforman en *arrays*, para que el programa las pueda leer y mostrar por pantalla. Estos *arrays* tienen una variable por cada píxel que tenga la imagen y cada uno de ellos contiene el color del que debe pintar ese píxel la pantalla para componer la imagen.

Como se muestra en la gráfica de la figura 4.16 se tarda menos en refrescar cuando cada rasgo es una imagen por separado que cuando la cara es una única imagen, ya que cuanto menor es la imagen menos tarda en refrescarla.

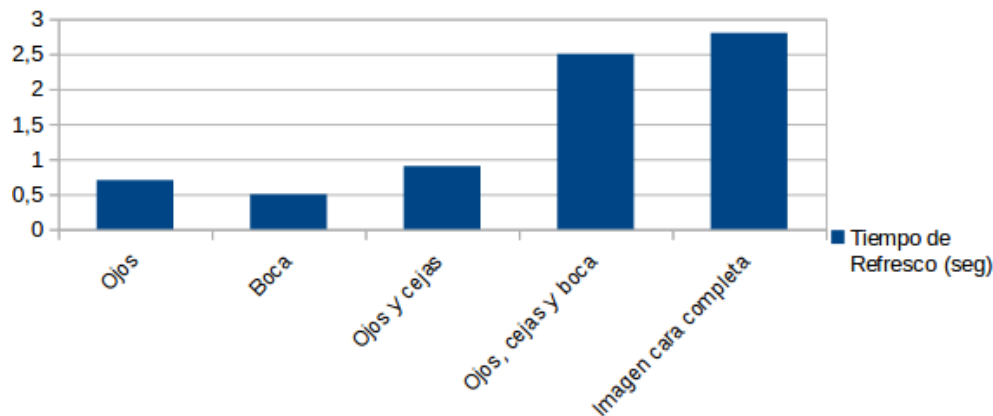


Fig. 4.16. Gráfica del tiempo de refresco de la imagen según le agregamos rasgos faciales

4.5.4. Implementación tarjeta SD

Para conseguir mostrar las imágenes por la pantalla utilizamos una tarjeta SD donde se guardan todas en formato .RAW. Al utilizar la tarjeta SD como fuente de imágenes es necesario incluir la siguiente librería:

- **Librería SdFat:** esta librería se basa en los sistemas de codificación de ficheros FAT16 y FAT 32. La librería SdFat proporciona una gama exhaustiva de funciones para el manejo de estas tarjetas de memoria.

La principal sentencia para acceder a estas imágenes es:

- **sd.begin(SD_CHIP_SELECT, SPI_FULL_SPEED):** siendo SD_CHIP_SELECT el pin 53 de nuestra pantalla y SPI_FULL_SPEED la máxima velocidad posible para acceder a estas imágenes.

Antes de inicializar la pantalla y tarjeta SD es importante establecer la velocidad de datos en bits por segundo (baudios) para la transmisión de datos en serie y comunicarse con el equipo. Esta herramienta nos permite saber si el programa está funcionando de manera correcta y si hemos accedido o no a la tarjeta SD.

4.5.5. Verificación código

Tras implementar el código para la muestra de imágenes, se realiza un test de prueba para comprobar que el *Subscriber* recoge los datos de manera adecuada. La sentencia que

utilizaremos para que el *Subscriber* suscriba al tema que se está publicando es:

```
ros :: Subscriber < geometry_msgs :: Twist > s("s", Callback_Velocidad);
```

Cuando se recibe un nuevo mensaje se llama a la función “ Callback_Velocidad”, con el mensaje como primer argumento. En dicha función es donde se elige las diferentes caras.

Lo primero que debemos hacer es verificar nuestro código del *Subscriber* en Arduino. Una vez verificado, se procede a cargarlo en nuestra placa. Por último, utilizando el terminal ejecutaremos el nodo ROS y, mediante las herramientas de línea de comandos rostopic se puede mostrar información de depuración sobre suscriptores. Las principales herramientas que utilizaremos son:

- **Rostopic echo <nombre del tema>** : esta herramienta muestra los mensajes publicados en un tema.
- **Rostopic list**: esta herramienta nos permite mostrar una lista de temas actuales.
- **Pub<topic-name><topic-type>[data...]**: esta herramienta es la más importante ya que con ella publicaremos los datos para el subcriber y así poder comprobar si funciona el programa Arduino correctamente, simulando que recibe datos de *Publisher* .

Gracias a este proceso se lleva a cabo las pruebas realizadas en la sección 6.

5. DISEÑO DE LA CARCASA DE LA CABEZA

Los diseños de la carcasa de la cabeza y de su base para colocarla en el robot son fundamentales, ya que tendrán el cometido de dar soporte a los diferentes componentes. En concreto, la carcasa de la cabeza da soporte y aloja en su interior la pantalla con el Arduino y el escudo LCD. Por otra parte, la base es la unión entre la cabeza y el tronco del robot. Dicho lo anterior, los diseños, además de estar pensados para la correcta integración de los elementos, se intentará que sean lo más estéticos posible.

En el desarrollo del capítulo se comenta la ingeniería de requerimientos para el diseño mecánico en la sección 5.1. A continuación, se explica brevemente el funcionamiento del programa de FREECAD (sección 5.2) para después comentar los diferentes diseños de la cabeza, teniendo en cuenta las proporciones y formas ideales del ser humano. Para estos diseños la cabeza se ha dividido en dos partes:

1. La carcasa de la cabeza en la sección 5.3.
2. La carcasa de la base en la sección 5.4.

Una vez tengamos las formas y medidas claras, se empieza a diseñar gráficamente las distintas partes que formen parte de dichas carcasas, para finalmente ensamblar en un diseño final (sección 5.5).

En el transcurso de este proceso se crean varias versiones hasta llegar a la más adecuada para nuestro objetivo.

5.1. Ingeniería de requerimientos para el desarrollo mecánico

La cabeza fue diseñada a partir de piezas producidas por impresión 3D, con la ayuda de la impresora Lion Pro 3D representada en la figura 5.1. Más adelante se explica el proceso de diseño con el programa FreeCad.



Fig. 5.1. Lion pro 3D

5.1.1. FREECAD

FReeCad es un programa de diseño gráfico asistido por ordenador. Está basado en Open CASCADE y programado en los lenguajes C++ y Python. Este programa utiliza técnicas de modelado paramétrico y está provisto de una arquitectura de *software* modular, permitiendo así, añadir de forma sencilla funcionalidades sin tener que cambiar el núcleo del sistema. Este programa constituye una herramienta muy potente para modelar y diseñar piezas en tres dimensiones.

El programa también ofrece la posibilidad de realizar ensamblajes y dibujos técnicos en dos dimensiones. Más adelante se ha explicado el funcionamiento de este programa y cómo se han diseñado las diferentes piezas del robot.

5.2. DISEÑO CON FREECAD

Las piezas de FreeCad se construyen mediante operaciones. Estas operaciones son el instrumento por el cual las piezas son creadas y modeladas. A través de ellas podemos crear sólidos, realizar cortes a un material, taladros, chaflanes etc.

Con las herramientas de las operaciones se pueden crear rectas, curvas y diferentes formas geométricas como círculos, elipses y demás figuras en dos dimensiones. Tras realizar esta figura en dos dimensiones, se puede pasar a tres dimensiones mediante diferentes operaciones como pueden ser: vaciado, revolución o extrusión introduciendo distintos parámetros relativos a ésta, como por ejemplo, el grosor de la pieza o la dirección en la

que se hace la operación.

Centrándonos en este trabajo, las diferentes piezas que se pueden encontrar más adelante se realizan gracias a la operación extruir y vaciado.

Por último, el entorno gráfico de FreeCad también ofrece la posibilidad de cambiar tanto la visión de la pieza como su perspectiva y así tener un mejor concepto de la pieza con la que se trabaja.

5.3. Diseño de la cabeza

En el diseño de la carcasa de la cabeza nos hemos intentado guiar por las proporciones y formas ideales del cuerpo humano, siendo, como se ha mencionado anteriormente, la cabeza del robot más grande que las proporciones ideales del cuerpo humano. Estas proporciones nos dicen que la cabeza debe medir aproximadamente una séptima parte de la altura total del individuo, por lo tanto, teniendo en cuenta lo anterior, y que nuestro robot medirá 150 cm de altura, con una simple división se concluye que la cabeza deberá medir aproximadamente unos 21 cm como máximo desde la barbilla hasta la parte más alta de la frente.

En cuanto a su forma geométrica, se ha escogido el rectángulo, tanto por su sencillez como por su similitud con la forma de la pantalla, que tiene unas dimensiones de 18x11 cm.

Respecto a los requisitos que debe cumplir el diseño, además de ser estético, tiene que estar preparado o incluir algún tipo de sistema de unión que permita ensamblar la cabeza con la base. También, como ya se ha mencionado, sirve de soporte para la pantalla, el Arduino y el escudo LCD.

La carcasa de la cabeza del robot es visual y estéticamente la cara del robot. En su diseño se incluye:

- El sistema de unión que ensamble la cabeza con la base.
- El soporte que sujeta los diferentes dispositivos mencionados en apartados anteriores.

De esta primera pieza se han creado dos versiones, dando lugar a dos modelos de carcasa de la cabeza distintos.

5.3.1. Cabeza versión 1

En esta primera versión, la cabeza está formada por dos piezas con la idea de que una haga la función de soporte y la otra de tapa para la pantalla. En la figura 5.2 a), se puede ver la parte trasera de la cabeza donde se alojarían los dispositivos que componen nuestro trabajo. Por otro lado, en la figura 5.2 b), se muestra como quedaría la carcasa de la cabeza en el caso de que se optase por esta opción, añadiéndole a la parte trasera una tapa con un ángulo de 45° .

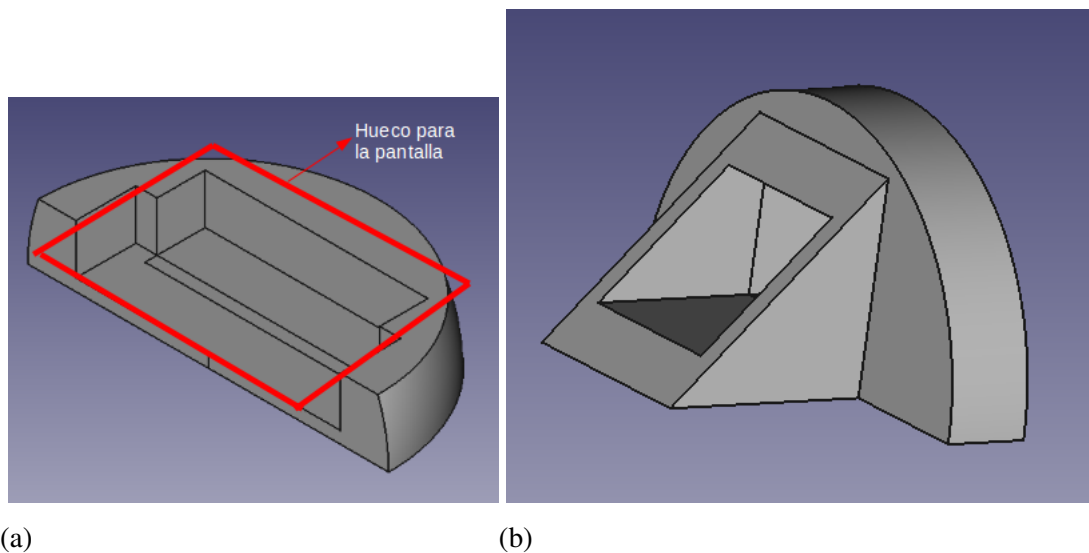


Fig. 5.2. Primera versión de la cabeza del robot.

Este diseño se descartó por diferentes razones:

- La complejidad que conllevaba realizar una tapa que cuadrara con esta carcasa cubriendo por completo todos los dispositivos, ya que la pantalla debe de estar en un ángulo de 45° para permitir que el usuario vea adecuadamente la cara del robot por la pantalla a una altura elevada.
- La altura y anchura de la pantalla fuerza a que el diámetro de la carcasa sea más grande que el espacio de la base donde irá colocada la cabeza del robot.

Esto se debe a que al tener la pantalla unas dimensiones de 18x11 cm, la semicircunferencia con la que se consigue esta figura debería de tener un radio mínimo de 22 cm, además, si le sumas el espacio que se necesita para la pared de la carcasa (la pared debe de tener un grosor mínimo de 1 cm) el radio se eleva a 24 cm. Esto se consigue gracias a las cuentas procedentes de la fórmula 5.1, la cual, añadiéndole las dimensiones de la pantalla (fórmula 5.2) nos da el resultado.

$$x^2 + y^2 = z^2 \quad (5.1)$$

$$18^2 + 11^2 = z^2 \quad (5.2)$$

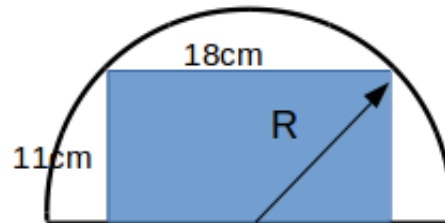


Fig. 5.3. Rectángulo circunscrito

A pesar de que la plataforma del robot tiene un diámetro de 34cm, al tener 12 taladros, que nos servirán para sujetar la cabeza, se debe de poner una carcasa para la cabeza de un tamaño menor o igual a 23cm.

5.3.2. Cabeza versión 2

Para simplificar el diseño hemos modificado la carcasa de la cabeza a una forma rectangular con los bordes redondeados. Esto permite alojar mejor los diferentes dispositivos. Como se puede ver en la figura 5.4 se ha creado un diseño teniendo en cuenta las medidas de la pantalla y los errores cometidos en el anterior diseño.

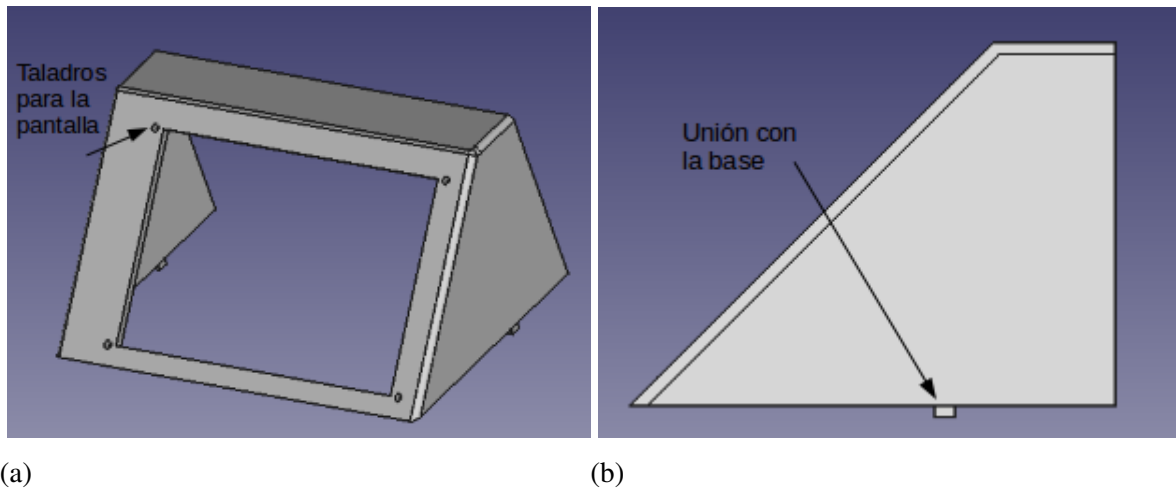


Fig. 5.4. Segunda versión de la cabeza del robot

Este diseño cumple los siguientes requisitos como se muestra en la figura 5.5:

- La anchura de la carcasa mide 23 cm para que, como se ha mencionado anteriormente, se pueda colocar entre los taladros de la base superior del robot (figura 5.5.a).
- La sujeción de la pantalla tiene un ángulo de 45° para que la persona con la que interactúe pueda visualizar mejor la imagen (figura 5.5.b).

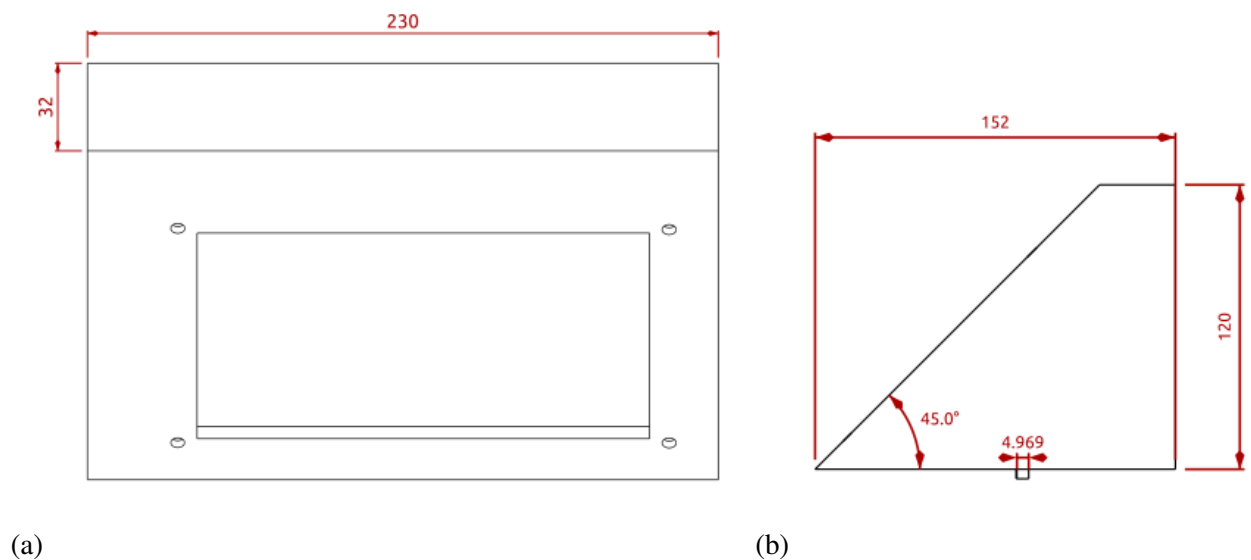


Fig. 5.5. Segunda versión de la cabeza del robot

Además, se han realizado 4 taladros para poder sujetar la pantalla y se han colocado dos uniones con la base en forma de pestaña cuadrada.

5.4. Diseño de la base

Para dar forma y entidad a la base de la cabeza del robot, se crea una pieza cuya geometría es cilíndrica para poder ajustarla a la la base superior del robot mediante 8 taladros de 3mm de diámetro. Sus dimensiones serán 1,5 cm de altura y 34 cm de diámetro. En la Figura 5.6 se puede apreciar las diferentes características mencionadas con sus medidas en las respectivas acotaciones:

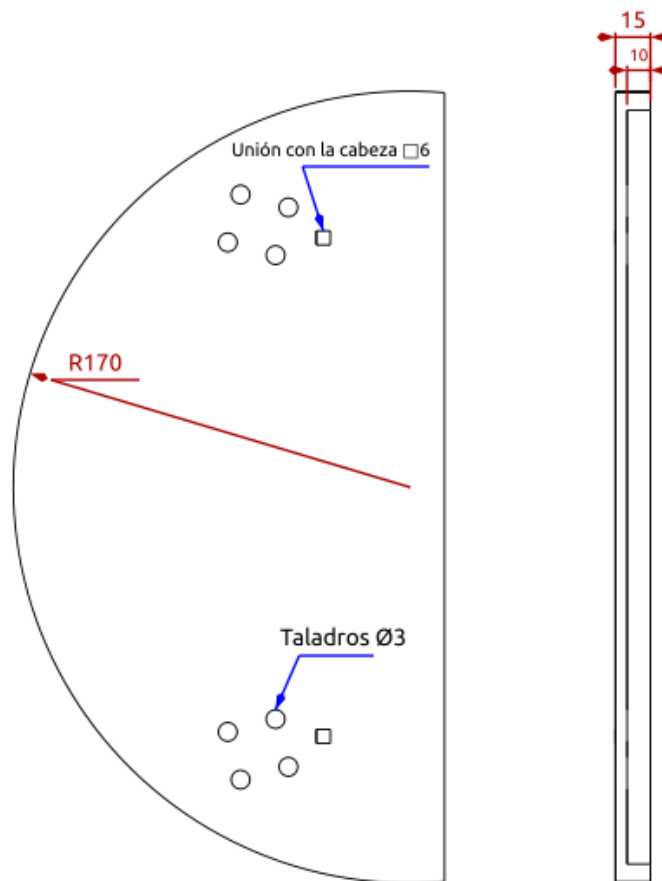


Fig. 5.6. Vistas de la base de la cabeza.

En la figura 5.7 se muestra el diseño de la base que, además de tener forma cilíndrica, se ha cortado por la mitad para disminuir el peso de la parte superior del robot. La base circular inferior ha sido vaciada para crear una unión con el robot más estética, en su base circular superior se ubica el punto de unión con la cabeza.

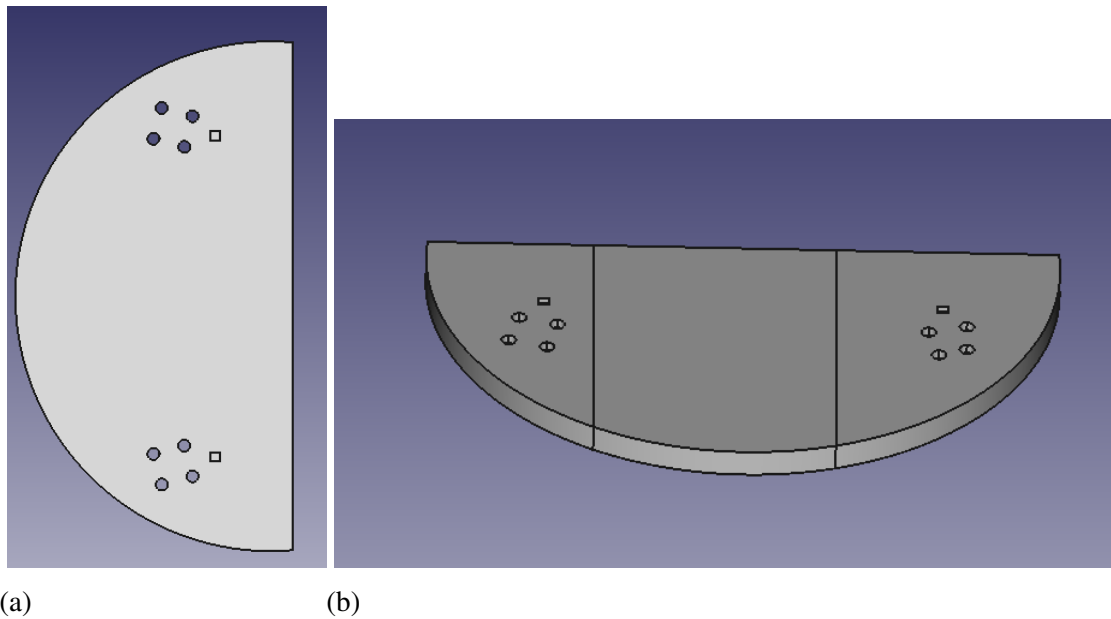


Fig. 5.7. Diseño de la base que une la cabeza con el cuerpo del robot.

5.5. Unión de los diseños

Una vez se han realizado y explicado los diseños de la carcasa de la cabeza y la base de ésta, es el momento de realizar su ensamblaje.

En la figura 5.8 se observa el diseño de las carcasas justo antes de ensamblarse, siguiendo la dirección del eje de unión y encajando las uniones en la forma de pestaña cuadrada anteriormente mencionadas.

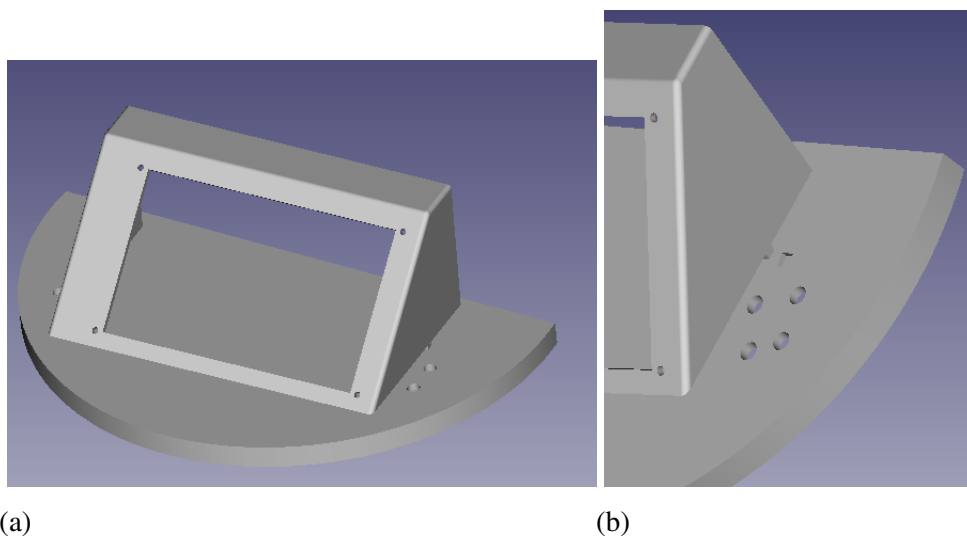


Fig. 5.8. Diseño de la cabeza del robot uniendo ambas partes.

6. CONSTRUCCIÓN Y PRUEBAS

El capítulo desarrolla el proceso de construcción de la cabeza del robot y los resultados de las pruebas de funcionamiento de la misma.

6.1. Construcción

Para iniciar la construcción se imprimen las piezas anteriormente diseñadas gracias a la impresora 3D Lion Pro. Para ello, se ha llevado a cabo una exportación de los diseños en FreeCAD a formato .stl y, posteriormente, se ha realizado su laminación mediante la aplicación de impresión 3D Repetier Host. De este modo, se ha generado el G-code de las piezas mediante la herramienta Slic3r de la aplicación. En este archivo se encuentran todas las instrucciones necesarias para que la impresora 3D pueda llevar a cabo la impresión de las piezas. En este punto cabe destacar que ha sido necesario llevar a cabo la impresión de la base de la cabeza en tres partes y la de la cabeza en dos, ya que el volumen de impresión no alcanza una anchura mayor de 19 cm. Tras ello, se han unido las partes entre sí con un pegamento instantáneo. En la figura 6.1 se muestra la base impresa y colocada sobre el cuerpo del robot mediante los 8 taladros mencionados en el diseño de la pieza.

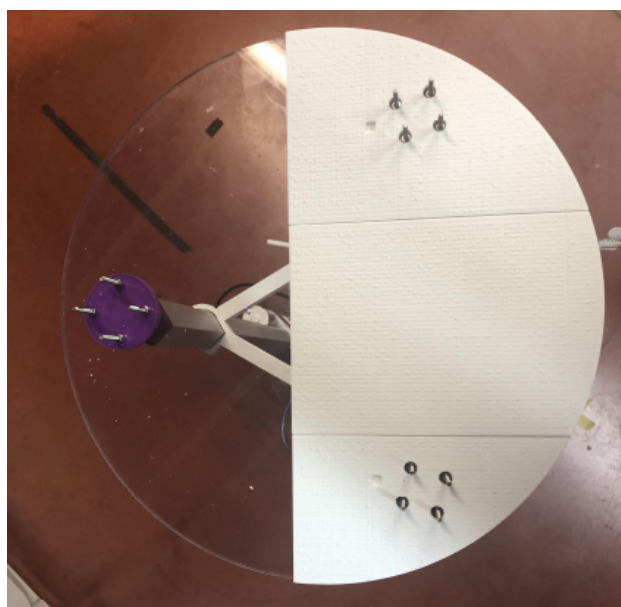


Fig. 6.1. Modelo impreso en 3D de la base.

Por otro lado, vemos en la figura 6.2 la carcasa de la cabeza impresa. En ella se pueden observar los taladros para la sujeción de la pantalla y las pestañas que se utilizan para la unión de la base con la cabeza. Como se puede ver en la figura 6.3, la cabeza se ha colocado entre los taladros de la base, esto se debe a un motivo meramente estético, ya que se pretende llevar a cabo los requisitos planteados en la sección 5.3.1 de la manera más acorde posible.

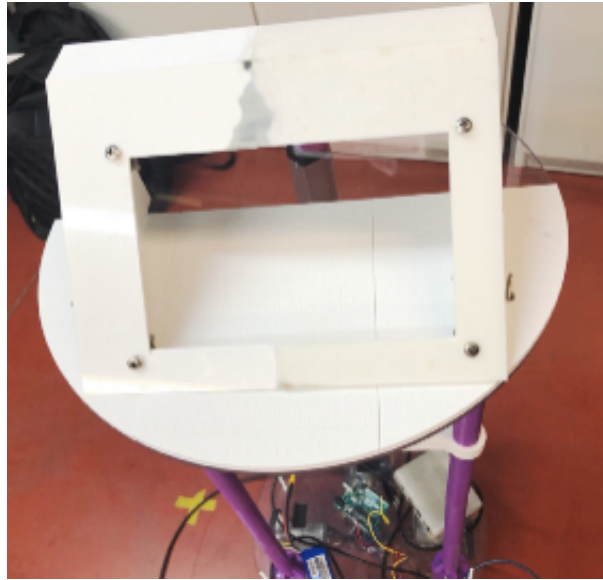


Fig. 6.2. Modelo impreso en 3D de la cabeza sobre la base.

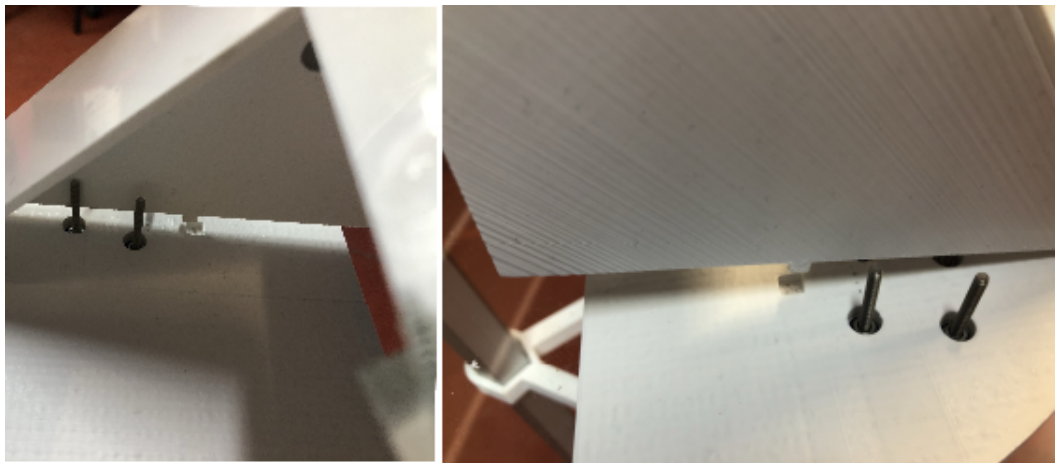


Fig. 6.3. Imagen en detalle de los taladros y la unión de la base con la cabeza.

Por último, se procede a pintar las piezas impresas en 3D para que queden de un color uniforme y conseguir una visión general del robot amigable. Además, tras llevar a cabo el montaje con todos los componentes explicados anteriormente en la sección 4.1 (ver figura

6.4) se ha completado el sistema de expresión visual sobre la plataforma móvil mob-E. En la figura 6.5 aparece el sistema montado y pintado.

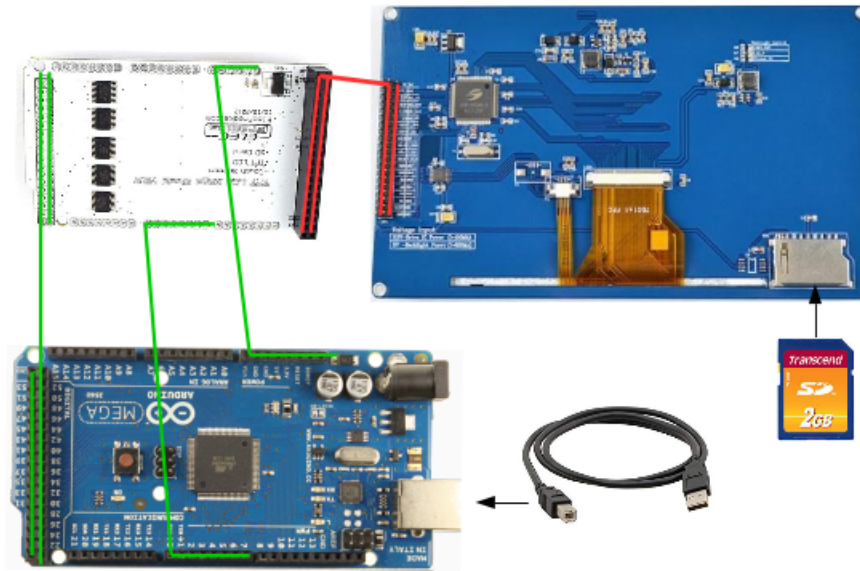


Fig. 6.4. Diagrama de conexión del sistema de expresión visual

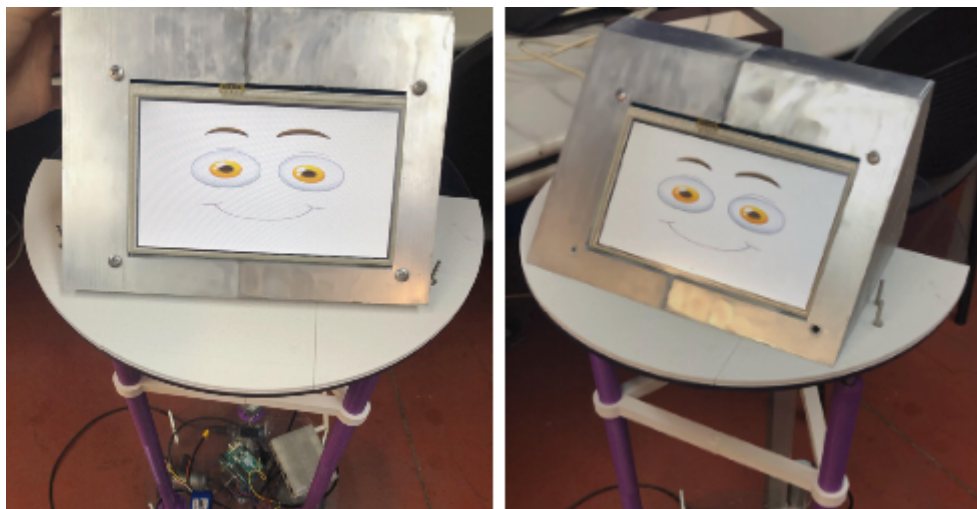


Fig. 6.5. Robot presentado en este trabajo tras llevar a cabo el montaje de los dispositivos electrónicos y pintado

6.2. Pruebas

En relación a las pruebas, cabe destacar que antes de ensamblar los elementos de la estructura interna del robot se verificó si los elementos fabricados cumplían las medidas requeridas por el diseño. Los resultados obtenidos fueron todos positivos.

Tras implementar todo el código de la expresión visual del robot, se realizaron las pruebas de funcionalidad. En cada parte de la funcionalidad desarrollada, se realizaron pruebas que permitieran determinar si realmente el código realizaba la función correctamente.

6.2.1. Prueba de reacción del sistema

Esta prueba se ha llevado a cabo mediante un pequeño programa en ROS sobre el sistema electrónico de control de la pantalla descrito en la sección 4.5. Gracias a este programa se consigue variar la velocidad para comprobar el tiempo que tarda el sistema en reaccionar desde que se le manda la orden de que el robot ha cambiado de velocidad hasta que la pantalla cambia de expresión. Se realiza la misma prueba para la variable booleana de llegada con éxito al destino.

Orden mandada	t 1 [s]	t 2 [s]	t 3 [s]	Media [s]
Velocidad = 0 m/s	0,21	0,22	0,21	0,21
Velocidad = 0,5 m/s	0,21	0,21	0,22	0,21
Destino = True	0,23	0,22	0,21	0,22
Destino = False	0,22	0,20	0,20	0,21

TABLA 6.1. TIEMPO EN REACCIONAR DEL SISTEMA

Como puede apreciarse en la tabla 6.1, los tiempos de reacción del sistema son muy similares para las diferentes variables. De modo que puede afirmarse que esta característica no será un punto crítico del sistema, ya que además se tiempo es lo suficientemente pequeño para que concuerde el estado interno del robot actual con la expresión que muestra el rostro del robot.

6.2.2. Prueba del estado de parpadeo

Para el estado de parpadeo se han probado ha realizar el movimiento a diferentes velocidades con el fin de encontrar un parpadeo lo más natural posible. La velocidad de parpadeo depende de dos características:

- El tamaño que usemos para la imagen de los ojos.
- El tiempo que pongamos de espera entre el tránsito de una imagen a otra.

En la tabla 6.2 se muestra las obtención de los tiempos al variar el tamaño de la imagen de los ojos y la valoración de la naturalidad que transmiten, tomando esta medida como una nota del 1 al 5 (siendo 1 la nota más baja y 5 la nota más alta). Es importante añadir que los tiempos están tomados como referencia desde el momento en el que se realiza la transición de ojos abiertos a cerrados, hasta que termina la transición de ojos cerrados a abiertos.

Píxeles de la imagen	t 1 [s]	t 2 [s]	t 3 [s]	Media [s]	Naturalidad
400x100	0,75	0,72	0,70	0,72	1
400x200	0,77	0,80	0,81	0,79	2
500x150	0,85	0,84	0,83	0,90	2
600x200	1,33	1,40	1,39	1,37	5
500x200	1,05	1,04	1,1	1,06	3

TABLA 6.2. TIEMPO DE PARPADEO Y NATURALIDAD QUE TRANSMITE

Como podemos observar en la tabla mencionada, no existe una dependencia lineal entre la velocidad de parpedo y la naturalidad que puede transmitir según el tamaño y el tiempo que tarda en el movimiento, por ello, se puede afirmar que el tamaño escogido en las secciones anteriores para los ojos(600x200 píxeles) es el más adecuado, ya que, a pesar de que el tiempo de parpadeo no es el mínimo, es un tiempo relativamente corto y el tamaño de los ojos esta acorde al resto de rasgos faciales.

En relación al tiempo de espera entre el parpadeo, se ha utilizado uno de 4,63 segundo, ya que, como se menciona en la sección 4.5.2, la secuencia de parpadeo se debe de produ-

cir 10 veces por minuto, con lo cual, mediante la ecuación 6.1 se consigue este resultado (en esta ecuación de resta al tiempo que debe de pasar entre cada parpadeo los segundos que tarda el robot en parpadear).

$$60/10 - 1,37 = 4,63 \quad (6.1)$$

Por otro lado, se ha comprobado que el sistema realice adecuadamente el tránsito del estado de parpadeo a dormido según el tiempo que el robot esté a una velocidad de 0 m/s. A partir de estos ensayos, se obtuvo un resultado satisfactorio ya que cumple con el tiempo especificado en la sección 4.5.2 de tres minutos.

6.2.3. Prueba de refresco de las diferentes expresiones

Tras comprobar el tiempo que tarda el sistema en reaccionar a las diferentes órdenes, se prueba el tiempo de refresco de las diferentes expresiones según el tamaño de la imagen, ya que como se ha comentado en la sección 4.5.3, el tiempo de refresco depende del número de píxeles de la imagen, por lo que se optó por realizar una transición de imágenes diferentes en cada cambio de estado. En las pruebas realizadas se obtuvieron los resultados que aparecen en la tabla 6.3 para las diferentes transiciones.

Transición de expresiones	t 1 [s]	t 2 [s]	t 3 [s]	Media [s]
Neutral-Feliz	1,02	0,95	0,99	0,98
Neutral-Triste	1,09	1,03	1,12	1,08
Neutral-parpadeo	1,37	1,38	1,41	1,38
Neutral-Asustado	1,09	1,21	1, 07	1,12
Asustado-Muy Asustado	0,91	1,02	0,99	0,97
Parpadeo-Dormido	0,52	0,59	0,58	0,56

TABLA 6.3. TIEMPO DE REFRESCO DE LAS DIFERENTES EXPRESIONES

En la tabla se observa lo comentado en la sección 4.5.3, la transición de expresiones depende de la cantidad de imágenes que se cambien y cuál es la imagen que se esta cambiando. La transición más corta es la del cambio de estado de parpadeo a dormido,

ya que únicamente se cambia la boca y se añade una imagen pequeña de Zs. Por otro lado, las transiciones más larga son: el cambio de estado de neutral al estado de parpadeo, debido a que el estado de parpadeo conlleva un movimiento más largo, y la transición de neutral a asustado, ya que se realiza un cambio de todos los rasgos faciales.

6.2.4. Evaluación expresiones

Dependiendo del estado interno del robot, la pantalla devuelve diferentes expresiones según la emoción que corresponda. En esta prueba se valoran las características indispensables estudiadas en el estado del arte, para que el rostro del robot muestre cercanía a la hora de realizar la interacción.

La expresión neutral (ver figura 6.6 es la que veremos al encender el robot. Esta expresión tiene las siguientes características: una cara sonriente, la separación de los ojos más grande que el radio de estos, las cejas se sitúan en una posición neutral y realiza el movimiento de parpadeo. Todo esto hace que sea acogida por el usuario de una manera natural, sin provocar miedo o confusión.

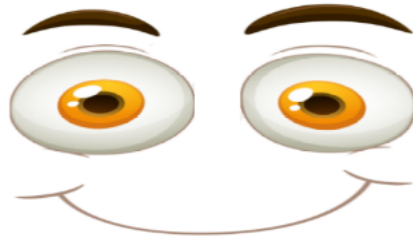


Fig. 6.6. Expresión del estado neutral.

Partiendo de esta expresión neutral podemos pasar a diferentes emociones:

- **Felicidad:** una expresión en la que el robot tiene una posición neutra de las cejas y destaca una amplia sonrisa. Entrando más en detalle, se muestran unas arrugas en las mejillas producto de la sonrisa.(Ver figura 6.7)



Fig. 6.7. Expresión del estado feliz.

- **Tristeza:** como se muestra en la figura 6.10 la boca se arquea hacia abajo. Las cejas están fruncidas, además de que los ojos miran hacia abajo cayendo el párpado superior. Por ultimo, se ha querido enfatizar la expresión añadiendo lágrimas.



Fig. 6.8. Expresión del estado triste.

- **Susto:** en esta expresión las cejas se arquean hacia arriba, ya que los ojos se abren ampliamente. Por otro lado, la boca se muestra apretada. Para la expresión del estado muy asustado todas estas características se enfatizan más, como se muestra en la figura 6.9.

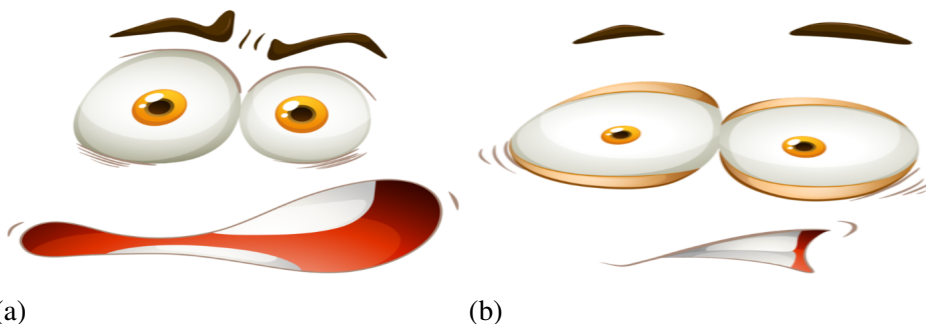


Fig. 6.9. a) Expresión del estado asustado. b) Expresión del estado muy asustado.

- **Dormido:** para la expresión de dormido, únicamente se han cerrado los ojos y se ha curvado la boca. Para no crear duda del estado del robot, se han añadidos los

símbolos de las Zs, ya que son los símbolos reconocibles de este estado.

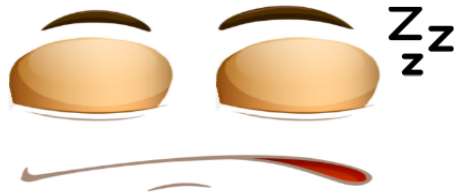


Fig. 6.10. Expresión del estado dormido.

7. DISCUSIÓN DE LOS RESULTADOS

En este capítulo se presentan las conclusiones del trabajo fin de grado, una vez finalizadas las pruebas, y se realiza una propuesta de trabajos futuros, que se consideran que podrían implementarse si se continuara con el desarrollo de este proyecto.

7.1. Conclusiones

El presente trabajo ha hecho que ponga en práctica muchos conocimientos adquiridos a lo largo de la carrera. No obstante, he tenido también que profundizar en distintos campos, tanto de la electrónica como de la programación. Esto me ha servido para aumentar el conocimiento de dichos estudios.

Para realizar las distintas expresiones del robot ha sido necesario el estudio y la reflexión en profundidad de todos los elementos que componen un robot social, con el fin de comprender las necesidades que éstos pueden tener. Por otro lado, antes de comenzar a utilizar las librerías, fue necesario documentarse sobre el funcionamiento y lenguaje utilizado por las herramientas de desarrollo de programación.

Cabe destacar que el esfuerzo invertido, tanto económico, como humano ha sido compensado por los resultados obtenidos. Esto es así debido a que el objetivo de este trabajo, el diseño y construcción de un sistema de expresividad para un robot móvil, se ha cumplido a un relativo bajo coste y mediante un mecanismo relativamente simple. Gracias a ello se han podido extraer numerosas conclusiones.

El robot de este trabajo responde a los principios de la robótica social. Por lo tanto, el robot es capaz de interactuar y comunicarse con los humanos. Esto lo realiza mediante gestos, conseguidos gracias a la transición de diferentes imágenes, pudiendo realizar cinco expresiones diferentes: feliz, triste, dormido, asustado y neutral. Todas estas emociones varían según el estado interno del robot, trabajando conjuntamente el cuerpo con la cabeza.

Tras haber finalizado el trabajo, podemos afirmar que hemos llevado a cabo un proceso correcto para conseguir unos objetivos adecuados, ya que finalmente se han conseguido.

do objetivos similares a los alcanzados en trabajos como el del Robot Maggie, un robot desarrollado por el Robotics Lab de la UC3M [48]. Ambos robots son móviles con capacidad de interacción. A pesar de su distinta dimensión se puede observar que su trabajo se aborda de la misma manera que en este trabajo el proceso de creación del sistema de interacción humano-robot.

El sistema diseñado se ha implementado y probado como se explica en la sección 6, siendo los resultados de su evaluación positivos. La cabeza es compatible con el resto de la arquitectura del cuerpo del robot y su funcionamiento es el esperado. Además, las pruebas demuestran que la implementación del trabajo funciona con una concordancia temporal adecuada.

A pesar de que el sistema funciona correctamente, cumpliendo los objetivos específicos, todavía existen una serie de inconvenientes que podrían solucionarse en trabajos futuros, centrados básicamente en el avance del programa implementado.

7.2. Trabajos futuros

Con el fin de dotar de una mayor funcionalidad a nuestro robot, y así tener una aplicación más completa, consideramos como líneas futuras la implementación de los siguientes aspectos que no llegamos a cubrir en nuestro trabajo:

- **Añadir el reconocimiento por voz:** la incorporación de un micrófono para la futura comunicación verbal con el robot, pudiendo realizar la implementación del reconocimiento por voz haciendo uso del sensor de voz *Elechouse Voice Recognition V3*. Gracias a él, se podría realizar un programa mediante la grabación de los comandos que queramos utilizar para que el robot reconozca cuando le llamen por su nombre o le realicen alguna orden y variar su estado de ánimo.
- **Mejorar la naturalidad en el rostro:** como se ha explicado en capítulos anteriores, la naturalidad a la hora de realizar una interacción humano-robot es muy importante. Como línea de mejora se propone reducir los tiempos de refresco de las imágenes, esto se podría realizar utilizando otra librería, la cual necesita otro tipo de tarjeta SD. Además, siguiendo con esta línea, se ha planteado la incorporación del movimiento de los ojos, según a donde se dirija el robot y del párpado inferior.

- **Mejorar el diseño de la cabeza:** llevar a cabo una mejora del diseño para que la cabeza esté preparada para poder realizar distintos movimientos. Un movimiento en el cual la cabeza gire sobre el cuello en cualquier sentido (girando sobre un eje vertical), y otro donde la cabeza y el cuello puedan inclinarse hacia adelante o hacia atrás (girando sobre un eje horizontal), según el estado de ánimo del robot o hacia dónde se dirija.
- **Prueba del robot en entornos con humanos:** ya que el objetivo final de este trabajo es la interacción humano-robot de un robot móvil pensado para ser un robot de servicios en un futuro, una prueba futura sería recoger las respuestas de distintos grupos de población ante la utilidad de este robot, analizando como responden a las diferentes expresiones. EL análisis se realizaría diferenciando entre distintos grupos de edad, diferencia de sexos y niveles de estudio.

8. ENTORNO SOCIO ECONÓMICO

En este capítulo se muestra el impacto social y económico esperado de la aplicación del resultado final del proyecto, y el presupuesto desglosado del coste total del proyecto.

8.1. Entorno socio económico

En relación al impacto socio-económico, en el mundo de la robótica se diferencian completamente estos dos impactos. Desde la perspectiva social, se puede observar cierta preocupación al verlos una amenaza de cara al mundo laboral. Sin embargo, el enfoque económico es bastante positivo, según el trabajo que realiza el profesor Salvador del Rey [49], el rendimiento de los robots va a aumentar en torno a un 5 %, al mismo tiempo que su coste se reducirá hasta un 20 % en el próximo año.

Nuestro robot es un robot móvil, con una plataforma con variación de altura y capaz de realizar una interacción humano-robot, pensado para pasar a ser un robot de servicios y poder realizar tareas de ayuda a los humanos, tanto en el ámbito de asistencia como en el de educación. Por ello, centraremos el análisis socio-económico en estas categorías.

Los robots sociales se están abriendo paso con éxito en el mundo de la asistencia terapéutica. Múltiples experimentos e investigaciones muestran que es beneficioso para la mejora de la calidad de vida de niños y mayores, en hospitales y residencias de ancianos, o para desarrollar las habilidades sociales en personas con trastornos del espectro autista. También puede facilitar la terapia de personas con alguna discapacidad mental o física. Por otro lado, la robótica social educativa es una herramienta para los profesores según Seldon [50], que afirma que este tipo de nuevas tecnologías ayudarán a crear un programa personalizado para el estudiante, el cual aprovechará mejor las clases con unos niveles de atención de casi el 100 % frente al 20-25 % actual.

Esta serie de ventajas se ven contrarrestadas por la opinión de la gente respecto a sus efectos sobre el mercado laboral [51]. En la actualidad, el 8 % de los puestos de trabajo son ocupados por robots, pero se espera que se eleve en 2020 este porcentaje al 26 %. Además, a pesar de que se pensaba que estos puestos de trabajo serían aquellos de baja

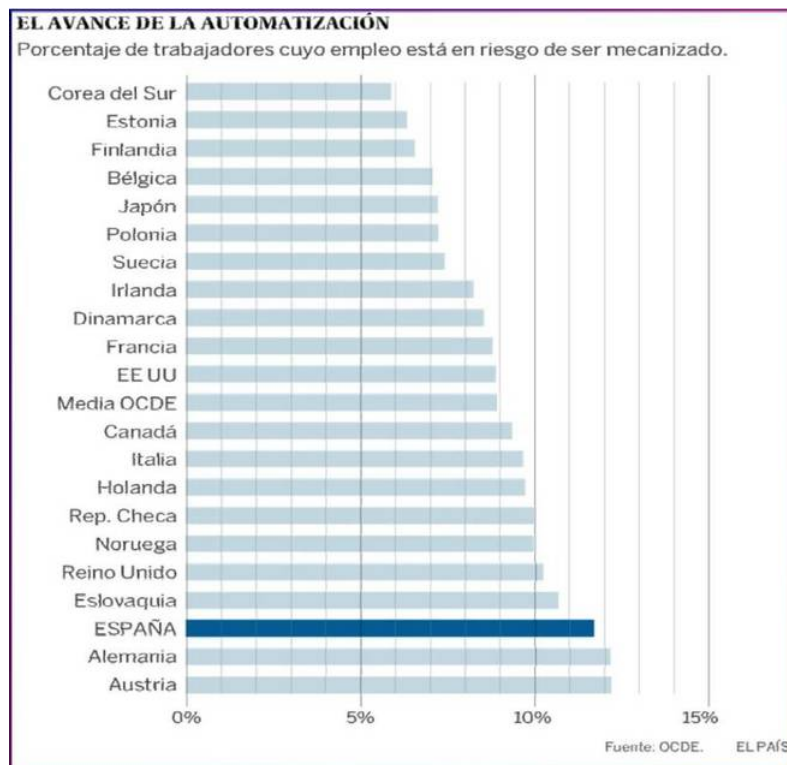


Fig. 8.1. Países con alto riesgo de sustitución de empleados por autómatas. Fuente: [52]

cualificación, los robots son capaces de sustituir también empleos de conocimientos intermedios como en la sanidad, la educación o tareas administrativas, ya que en ella se realizan numerosas tareas repetitivas y fáciles de automatizar. En la figura 8.1 podemos comparar entre los países de la OCDE el porcentaje de trabajadores cuyo puesto de trabajo se encuentra en riesgo de ser sustituido por un autómatas.

Por otro lado, en relación con el entorno económico, además de los datos mencionados anteriormente, según el profesor Salvador del Rey, mientras que las personas doblan su productividad cada diez años, los robots, como mínimo, lo hacen cada cuatro. Sin olvidar que el tiempo de amortización de un robot era en 2015 de 5,3 años y en 2025 se reducirá a 1,3 años. Lo cual demuestra que existe un entorno económico positivo en el mundo de la robótica.

Cabe destacar, centrándonos en nuestro proyecto, que la inclusión de la parte social en un robot no implica un mayor coste adicional a la fabricación, y hace que sea más atractivo a ojos humanos, lo que puede llegar a suponer un aumento en las ventas del robot.

Para finalizar el análisis, añadiremos una breve mención al impacto ambiental. A pesar

de que tiene poca importancia, debido a tratarse de un desarrollo digital, el consumo de electricidad es elevado, tanto a la hora de realizar las investigaciones como durante la utilización del robot, por lo que se podría considerar un impacto negativo hacia el medio ambiente.

8.2. Presupuesto

A continuación, se muestra el presupuesto del coste del proyecto, dividido en costes de personal, costes de hardware y costes de software.

CÓDIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO	PRECIO TOTAL
Capítulo 1: Personal					
1.01		ANÁLISIS			
	h	Estudio de las herramientas a utilizar y preparación del proyecto	200	10,00 €	2.000,00 €
1.02		DISEÑO			
	h	Diseño y programación del sistema de vivacidad	200	10,00 €	2.000,00 €
1.03		PRUEBAS			
	h	Evaluación del sistema implementado en el robot	100	10,00 €	1.000,00 €
		DOCUMENTACIÓN			
	h	Redacción del documento sobre el sistema propuesto	180	10,00 €	1.800,00 €
		TUTORÍAS			
1.04	h		50	15,00 €	750,00 €
Subtotal					7.550,00 €

Fig. 8.2. Costes de personal

CÓDIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO	PRECIO TOTAL
Capítulo 2: Software Expresión Visual					
2.01		SISTEMA OPERATIVO			
	Ud	Sistema operativo Ubuntu		0,00 €	0,00 €
2.02		ROS			
	Ud	Framework ROS para el desarrollo de aplicaciones robóticas		0,00 €	0,00 €
2.03		COMPILADOR			
	Ud	Compilador Arduino		0,00 €	0,00 €
2.04		ENTORNO DE DESARROLLO			
	Ud	Entorno de desarrollo FreeCAD		0,00 €	0,00 €
2.05		PAQUETE OFFICE			
	Ud	Microsoft Office Word y Microsoft Office Excel		0,00 €	0,00 €
Subtotal					0,00 €

Fig. 8.3. Costes de software

CÓDIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO	PRECIO TOTAL
Capítulo 3: Hardware					
3.01		Requerimientos para el desarrollo electrónico			
	Ud	Arduino Mega 2560	1	13,99 €	13,99 €
	Ud	Módulo de Pantalla Táctil para Arduino AVR STM32 ARM SSD1963	1	58,29 €	58,29 €
	Ud	TFT LCD Mega Shield V2.2 Tarjeta de expansión para pantalla táctil	1	10,80 €	10,80 €
	Ud	Tarjeta de Memoria SD de 2 GB	1	8,25 €	8,25 €
3.02		ORDENADOR			
	Ud	Ordenador portátil toshiba satellite Intel Core i7	1	535,00 €	535,00 €
3.03		Requerimientos para el desarrollo mecánico			
	Ud	Impresora LION PRO 3D	1	1.147,50 €	1.147,50 €
		PLA INGEO Blanco 1.75mm	1	17,95 €	17,95 €
3.04		ROBOT Mob-e			
	Ud	Radxa Rock Pro SBC. Supplied with wiring and box	1	100,00 €	100,00 €
	Ud	Arduino UNO	1	25,00 €	25,00 €
	Ud	Pololu 1442 gear motor	2	35,00 €	70,00 €
	Ud	Pololu set of 2 wheels	1	8,00 €	8,00 €
	Ud	Pololu set of 2 mounting hubs	1	6,00 €	6,00 €
	Ud	Digilent dual motor driver	1	6,00 €	6,00 €
	Ud	Cluster wheel	2	3,00 €	6,00 €
	Ud	LiPo battery	1	20,00 €	20,00 €
	Ud	External USB battery	1	20,00 €	20,00 €
	Ud	Asus Xtion camera	1	120,00 €	120,00 €
Subtotal					2.172,78 €

Fig. 8.4. Costes de hardware

Por último, se presenta el coste total del proyecto teniendo en cuenta todos los gastos.

RESUMEN PRESUPUESTO	
CAPÍTULO 1: COSTES DE PERSONAL	7.550,00 €
CAPÍTULO 2: COSTES DE HARDWARE	2.172,78 €
CAPÍTULO 3: COSTES DE SOFTWARE	0,00 €
TOTAL	9.722,78 €

Fig. 8.5. Costes total

9. METODOLOGÍA Y PLAN DE TRABAJO

Se presenta la planificación mediante un diagrama de Gantt, que muestra gráficamente el cronograma del proyecto, indicando la duración de cada proceso realizado (ver figura 9.1).

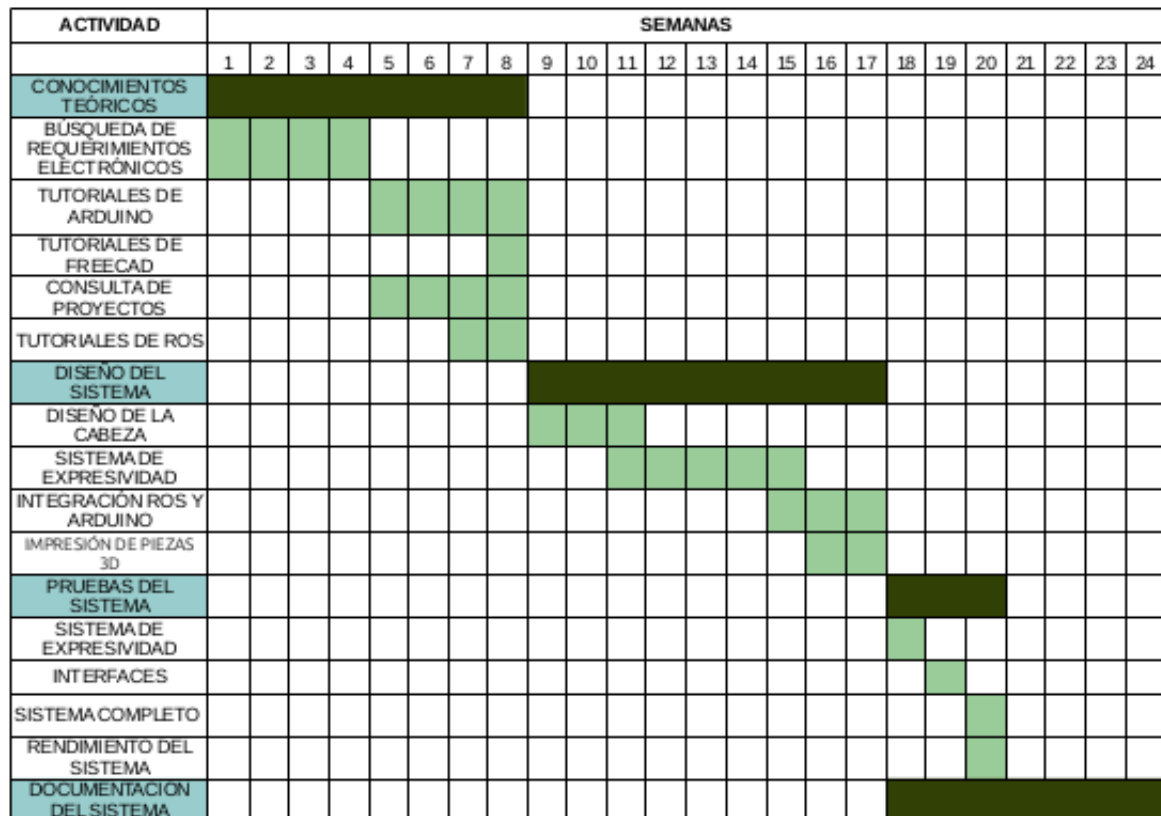


Fig. 9.1. Diagrama de Gantt del proyecto

BIBLIOGRAFÍA

- [1] *Definición de robótica*, R.A.E. de la Lengua. 2018. [En línea] Recuperado de: <https://dle.rae.es/srv/fetch?id=WYTm4uf>. Última visita: septiembre 2019.
- [2] S. Moriello, *Robots sociales, la nueva generación*, Megatendencias. [En línea]. Recuperado de: https://www.tendencias21.net/Robots-sociales-la-nueva-generacion_a2833.html. Última visita: septiembre 2019.
- [3] A. Vidal, Á. Castro-González, F. Alonso Martín, J. Castillo y M. Salichs, “Evolución de la robótica social y nuevas tendencias”, sep. de 2017.
- [4] P. E. y W. V. Friesen, “Facial Action Coding System: A Technique for the Measurement of Facial Movement”, *Consulting Psychologists Press, Palo Alto*, 1978.
- [5] R. A. Hinde, “Non Verbal Communication.”, *Cambridge*, 1972.
- [6] R. W. Picard, *Affective Computing*. Cambridge, MA, USA: MIT Press, 1997.
- [7] *Universidad Carlos III de Madrid*, «Trabajo de Fin de Grado UC3M: Evitar el plagio», Biblioteca UC3M, 2019. [En línea] Recuperado de: <http://uc3m.libguides.com/TFG/plagio>. Última visita: septiembre 2019.
- [8] M. J. S. González, “Regulación legal de la robótica y la inteligencia artificial: Retos de futuro.”, *Revista Juridica de la Universidad de León*, 4, 2017.
- [9] *GNU General Public License*, Wikipedia, 2018. [En línea]. Recuperado de: https://es.wikipedia.org/wiki/GNU_General_Public_License.. Última visita: septiembre 2019.
- [10] *Sistema Operativo Robótico*, Wikipedia, 2018. [En línea] Recuperado de: https://es.wikipedia.org/wiki/Sistema_Operativo_Robótico. Última visita: septiembre 2019.
- [11] *Licencia BSD*, Wikipedia, 2018. [En línea] Recuperado de: https://es.wikipedia.org/wiki/Licencia_BSD. Última visita: septiembre 2019.
- [12] *Introduccion Arduino*, Arduino. [En línea] Recuperado de: <https://www.arduino.cc/en/guide/introduction>. Última visita: septiembre 2019.

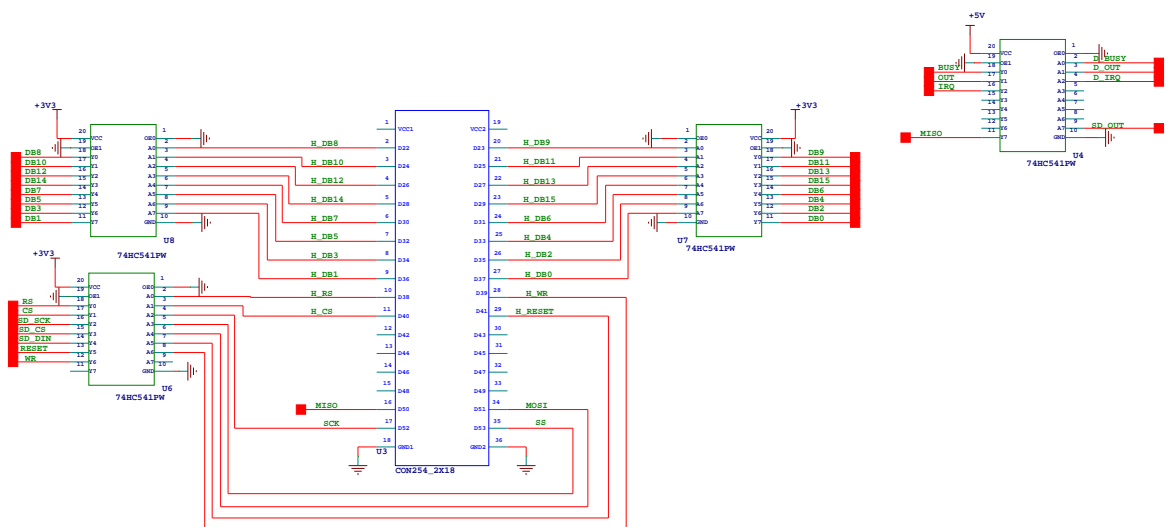
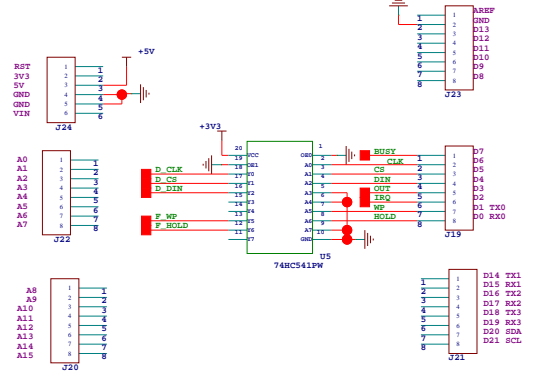
- [13] *Aspectos éticos*, Universidad Carlos III de Madrid, 2017. [En línea] Recuperado de: https://www.uc3m.es/ss/Satellite/LogoHRS4R/es/TextoMixta/1371234170165/Aspectos_eticos. Última visita: septiembre 2019.
- [14] M. Mori, “The Uncanny Valley. Energy”, 7(4), *pages 33-35*, 1970.
- [15] K. Dautenhahn y A. Billard, “Bringing up Robots or - The Psychology of Socially Intelligent Robots: From Theory to Implementation”, en *Proc. 3rd International Conference on Autonomous Agents*, 1999, pp. 366-367.
- [16] B. Duffy, “The Social Robot”, *Tesis doctoral, Univ. College Dublin*, 2000.
- [17] T. Fong, I. Nourbakhsh y K. Dautenhahn, *A Survey of Socially Interactive Robots*, 2002.
- [18] W. G. W. (s.f.), [En línea] Recuperado de: https://es.wikipedia.org/wiki/William_Grey_Walter. Última visita: septiembre 2019.
- [19] T. R. Journal, [En línea] Recuperado de: <http://www.rutherfordjournal.org/article020101.html>. Última visita: septiembre 2019.
- [20] B. R. Laboratory, [En línea] Recuperado de: <http://www.bristol.ac.uk/news/2008/212017945378.html>. Última visita: septiembre 2019.
- [21] C. Breazeal, “Towards Sociable Robots”, *ROBOTICS AND AUTONOMOUS SYSTEMS*, vol. 42, pp. 167-175, 2003.
- [22] D. Parisi, *Future robots*. USA: John Benjamins Publishing Company, 2014.
- [23] D. Rodríguez-Losada et al., “Urbano, an Interactive Mobile Tour-Guide Robot”, en. jul. de 2008.
- [24] D. R.-L. González, “Slam geométrico en tiempo real para robots móviles en interiores basado en ekf”, 2004.
- [25] R. URBANO, [En línea] Recuperado de: <https://blogs.upm.es/controlinteligente/en/gallery/>. Última visita: septiembre 2019.
- [26] F. Kaplan, “Everyday robotics: robots as everyday objects”, *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, 2005.

- [27] G. F. Melson et al., “Robots as dogs?: children’s interactions with the robotic dog AIBO and a live Australian shepherd”, en *In SIGCHI Conference on Human Factors in Computing Systems*, 2005, pp. 1649-1652.
- [28] *Sony Aibo*, [En línea] Recuperado de: <https://clipset.20minutos.es/este-es-el-nuevo-sony-aibo-2018/>. Última visita: septiembre 2019.
- [29] *Beatbots*, «MyKeepon,» 2011. [En línea] Recuperado de: <https://robots.ieee.org/robots/keepon/>. Última visita: septiembre 2019.
- [30] B. Gates, “Un robot en cada casa”, *Revista Investigación y Ciencia*, marzo, N 366, p. 38-45. Barcelona, Editorial Prensa Científica., 2007.
- [31] B. Graf, C. Parlitz y M. Hägele, “Robotic Home Assistant Care-O-bot”, ene. de 2009, pp. 312-320.
- [32] D. A. B. L. y Sergio Martini Popoli, “? Qué hay más allá del Valle Inquietante ?”, Robótica y Cibernética(ROBCIB), Madrid, 2014.
- [33] *Cuen, D.* (20 de Julio de 2011), En el inquietante valle de robots. [En línea] Recuperado de: http://www.bbc.co.uk/blogs/mundo/un_mundo_feliz/2011/07/en_el_inquietante_valle_de_rob.html. Última visita: septiembre 2019.
- [34] C. F. DiSalvo, F. Gemperle, J. Forlizzi y S. B. Kiesler, “All robots are not created equal: the design and perception of humanoid robot heads”, en *Symposium on Designing Interactive Systems*, 2002.
- [35] P. Ekman, “Emotion in the Human Face.”, *Cambridge University Press, Cambridge UK.*, 1982.
- [36] K. MacDorman y H. Ishiguro, “The uncanny advantage of using androids in cognitive science research”, *Interaction Studies*, vol. 7, pp. 297-337, ene. de 2006. doi: [10.1075/is.7.3.03mac](https://doi.org/10.1075/is.7.3.03mac).
- [37] *Robot Qrio*, [En línea] Recuperado de: <https://robots.ieee.org/robots/qrio/>. Última visita: septiembre 2019.
- [38] R. A. P. K. Dick, [En línea] Recuperado de: <https://www.hansonrobotics.com/philip-k-dick/>. Última visita: septiembre 2019.

- [39] G. Mandler, *The Psychology of Facial Expression*, J. A. Russell y J. M. Fernández-Dols, eds., ép. Studies in Emotion and Social Interaction. Cambridge University Press, 1997. doi: [10.1017/CB09780511659911](https://doi.org/10.1017/CB09780511659911).
- [40] t. D. Evans, “Can robots have emotions?”, en *Emotion: The Science of Sentiment*. Oxford University, 2001, cap. 5.
- [41] P. Ekman, “Universals and cultural differences in facial expressions of emotion.”, *Nebraska Symposium on Motivation.*, pp. 207-283, 1972.
- [42] *Rostro genérico para máquinas que interactúan con personas - Scientific Figure on ResearchGate*, [En línea] Recuperado de: https://www.researchgate.net/figure/Figura-2-Expresiones-faciales-universales-propuestas-por-Ekman-1972_fig1_321901828. Última visita: septiembre 2019.
- [43] M. ARGYLE, *Psicología del comportamiento interpersonal*. Alianza, 1978.
- [44] *Manual de usuario PANTALLA TFT 4,3"480X272 TACTIL SSD1963*, [En línea] Recuperado de: https://www.electronicaembajadores.com/datos/manuales/lc/lctf/LCTFT43_manual.pdf. Última visita: septiembre 2019.
- [45] *ARDUINO*, ARDUINO Language Reference. Recuperado de: <http://robotic.media.mit.edu/wp-content/uploads/sites/14/2015/01/Breazeal-RAS-03.pdf>. Última visita: septiembre 2019.
- [46] *Ejemplo de red P2P*, Language Reference. [En línea] Recuperado de: <https://www.xatakamovil.com/conectividad/metodos-para-compartir-archivos-y-contenidos-en-internet-iiip2p-el-autentico-espiritu-del-file-sharing>. Última visita: septiembre 2019.
- [47] M. J Doughty, “Further Assessment of Gender- and Blink Pattern-Related Differences in the Spontaneous Eyeblink Activity in Primary Gaze in Young Adult Humans”, *Optometry and vision science : official publication of the American Academy of Optometry*, vol. 79, pp. 439-47, jul. de 2002.
- [48] M. Salichs et al., “Maggie: A Robotic Platform for Human-Robot Social Interaction”, jul. de 2006, pp. 1-7.

- [49] *Publicación del profesor Salvador del Rey, catedrático de Derecho*, [En línea] Recuperado de: https://www.cuatrecasas.com/es/actualidad/notas_prensa/nuevos_retos_en_el_marco_regulatorio_de_los_recursos_humanos.html. Última visita: septiembre 2019.
- [50] S. A. Seldon y O. Adiboye, *The Fourth Education Revolution: Will Artificial Intelligence Liberate or Infantilise Humanity*. University of Buckingham Press, 2018.
- [51] M. Rius, “Asistidos por autómatas.”, *La Vanguardia*, 2016.
- [52] *Sánchez, C. (15 de 08 de 2016). El uso de robots se acelera y amenaza con destruir decenas de miles de empleos. El Confidencial*, [En línea] Recuperado de: https://www.elconfidencial.com/economia/2016-08-15/robots-automatizacion-ccoo-empleo-maquinas-industria_1245720/. Última visita: septiembre 2019.

ANEXO A: PINEADO DE LA TARJETA DE EXPANSIÓN ESCUDO LCD TÁCTIL PARA ARDUINO MEGA



ANEXO B: MANUAL PANTALLA TFT LCD SSD1963

7 inch TFT LCD Module User Guide

Rev 1.0, June 2011



<http://stores.ebay.com/Ego-China-Electronics>



**7" TFT LCD Module with Touch Panel
SSD1963 Controller**

Table of Contents

1 Introduction.....	2
2 Features	3
3 Hardware Connection	4
3.1 module output to display panel	4
3.2 module inputs from user MPU interface	5
4 Mechanical dimension	6

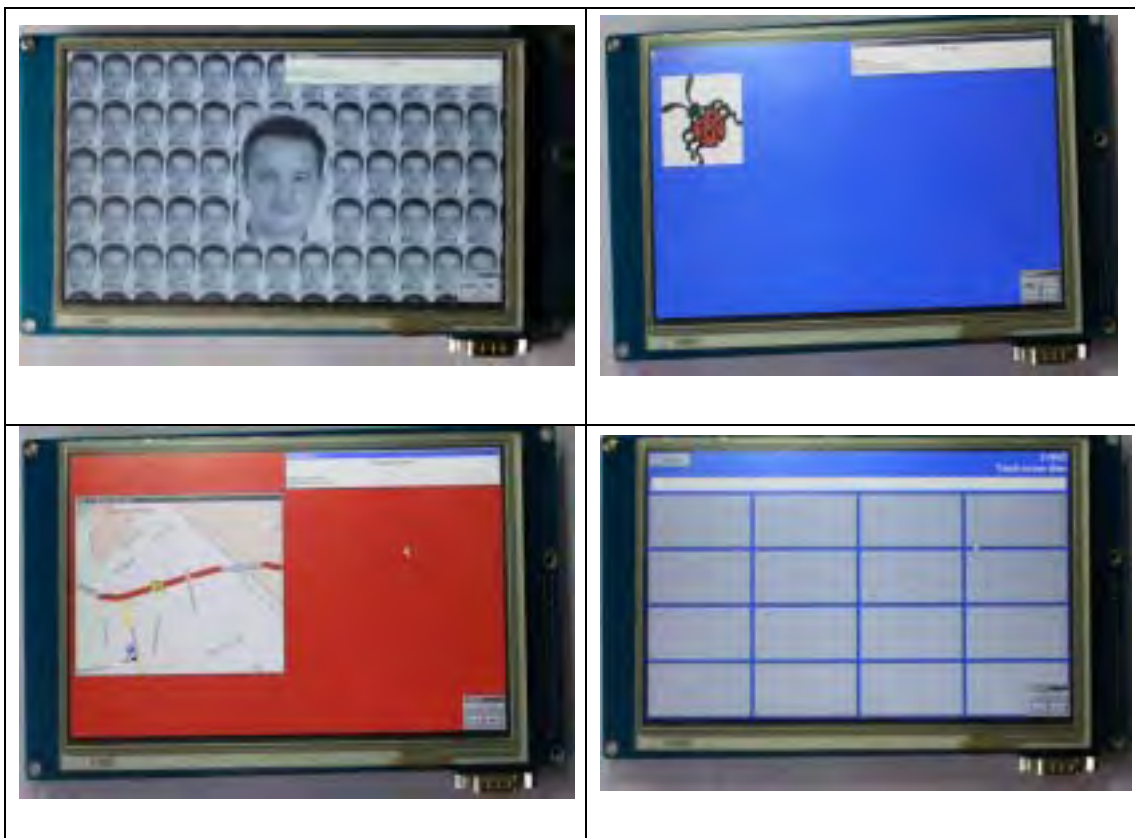
1 Introduction

The 7 inch TFT LCD module is 800x480 resolution with RGB666 which provide 64K colors. The module integrate a SSD1963 LCD controller which simplifier the user interface , graphic memory management and special effects to LCD panel. The module can be easily be controlled by any MCUs , ARM , DSP , FPGA and etc via the parallel bus. It also can be used to directly connect to Mini-STM32 develop board which can be found in our eBay store, demonstration C source projects is based on this develop board.



2 Features

- Display resolution: 800 x 480
- Display mode: 18bit RGB666
- Graphic memory: 1215K byte
- 16bits parallel user interface
- Hardware rotation of 0, 90, 180, 270 degree
- Hardware display mirroring
- Hardware windowing
- Dynamic Backlight Control (DBC) via PWM signal
- Programmable brightness, contrast and saturation control



3 Hardware Connection

3.1 module output to display panel

Pin	Signals	I/O	Function Description
1	VLED+	Power	Power for LED backl ight (Anode)
2	VLED+	Power	Power for LED backl ight (Anode)
3	VLED-	Power	Power for LED backlight (Cathode)
4	VLED-	Power	Power for LED backlight (Cathode)
5	GND	Power	Power Ground
6	VCOM	Input	Common voltage
7	DVDD	Input	Power for Digital Circuit
8	MODE	Input	DE/SYNC mode select
9	DE	Input	Data Input Enable
10	VS	Input	Vertical Sync Input
11	HS	Input	Horizontal Sync Input
12-19	B7-B0	Input	Blue data
20-27	G7-G0	Input	Green data
28-35	R7-R0	Input	Red data
36	GND	Power	Power Ground
37	DCLK	Input	Sample clock
38	GND	Power	Power Ground
39	L/R	Input	Left / right selection
40	U/D	Input	Up/down selection
41	VGH	Power	Gate ON Voltage
42	VGL	Power	Gate OFF Voltage
43	AVDD	Power	Power for Analog Circuit
44	RESET	Input	Global reset pin.
45	NC		No connection
46	VCOM	Input	Common Voltage
47	DI THB	Input	Dithering function
48	GND	Power	Power Ground
49	NC		No connection
50	NC		No connection

3.2 module inputs from user MPU interface

Pin	Signals	External Connection	Function Description
1	3.3V	Power Supply	Power supply for LCD and logic (3.3V)
2	SCK	TP	Touch panel SPI SCK
3,33,35,36	GND	Power Supply	GND
4	CS	TP	Touch panel SPI CS
5	RST	MPU	Active LOW Reset signal
6	MOSI	TP	Touch panel SPI MOSI
7	RD	MPU	8080 MPU Read Signal active LOW
8	MISO	TP	Touch panel SPI MISO
9	WR	MPU	8080 MPU Write Signal active LOW
10	IRQn	TP	Touch panel SPI IRQn
11	CS	MPU	Active LOW Chip Select signal
12	NC		No Connect
13	RS	MPU	Register Select. RS=1: Command, RS=0: Data
14	NC		No Connect
15	D7	MPU	16 bit bidirectional data bus
16	NC		No Connect
17	D6	MPU	16 bit bidirectional data bus
18	NC		No Connect
19	D3	MPU	16 bit bidirectional data bus
20	D13	MPU	16 bit bidirectional data bus
21	D5	MPU	16 bit bidirectional data bus
22	D12	MPU	16 bit bidirectional data bus
23	D4	MPU	16 bit bidirectional data bus
24,25	NC		No Connect
26	D11	MPU	16 bit bidirectional data bus
27	D2	MPU	16 bit bidirectional data bus
28	D10	MPU	16 bit bidirectional data bus
29	D1	MPU	16 bit bidirectional data bus
30	D9	MPU	16 bit bidirectional data bus
31	D0	MPU	16 bit bidirectional data bus
32	D14	MPU	16 bit bidirectional data bus
34	D8	MPU	16 bit bidirectional data bus
38	BK_PWM	GPIO	PWM input for backlight
37,39	5V	Power Supply	Power supply for LCD (5V)
40	D15	MPU	16 bit bidirectional data bus

4 Mechanical dimension

Note: All dimensions are calculated by mm unit

